ELECTRONIC LIBRARY

1. ADMINISTRATIVE APPLICATIONS
2. OVERVIEW OF NETWORK AND DESKTOP ENVIRONMENT
3. LAN/WAN DIAGRAM
4. GENERAL SERVER INOFRMATION
5. FIPS PUBLICATION 106
6. HELP DESK CALL AVERAGES
7. SAMPLE MONTHLY REPORT 1
8. SAMPLE MONTHLY REPORT 2
9. SAMPLE MONTHLY REPORT 3
10. SAMPLE SOP TEMPLATE

Volpe Center Administrative Applications

1 System Development Overview

The day-to-day system development operations of the Volpe Center are supported by a number of mission-critical applications. Most of these applications are owned and used primarily by the various Divisions within the Office of Administrative Services including Financial Management, Acquisition, Facilities Management, Human Resources, and Administrative Services. Some, however, are in general use throughout the Center. All were developed (or acquired) and are now maintained and operated by the Volpe Computer Center and their support contractor.

There are 26 administrative systems that could be termed legacy. Legacy systems were originally developed primarily in a proprietary Compaq (formerly Digital Equipment Corporation (DEC)) environment. They were built using COBOL and S1032, the premier database management system (DBMS) from the old DEC environment, and run under the OpenVMS operating system. A few of the legacy systems were built using HOLOS and Clipper. These systems are being scheduled for replacement but until such time are still being maintained and updated. They are currently still critical to the Volpe Center mission and a significant element in the day-to-day workload of the support contractor.

The current direction for new administrative applications is to acquire commercially-available solutions. The PRISM system has been acquired and has been implemented to support the Acquisition function. The Center is part of a DOT-wide effort called Delphi to implement Oracle Financials for Financial Management. Other packages are being investigated for other functional areas. The intent is to integrate these individual packages using data warehouse technology. The Volpe Intranet will serve as the means of making information available. The UNIX operating system and the ORACLE DBMS will be the platform for all applications.

The Computer Center's support contractor maintains the legacy applications making changes whenever new requirements are identified. The contractor also provides day-to-day production support for all applications currently in use. Finally, the contractor provides technical assessment of commercial packages and analysis, integration, design, coding, testing, documentation, and training associated with implementing any such package.

Descriptions of the following administrative applications are provided as listed below and grouped by functional business areas

Finance Applications

1. DAFIS Interface
2. Financial Status of Programs (FSOP)
3. Funds Certification
4. Plan
5. Total Cost
6. Work Plan Budget (WPB)
7. Invoice Tracking

8. Labor
9. Payroll
10. IEC Delphi

Acquisition

1. Contract Administration System (CAS)
2. Procurement Information Management (PRIM)
3. PRISM

Human Resources

1. Personnel
2. Full Time Equivalency (FTE)
3. Minority

Facilities

1. Property
2. Communication
3. Supply

Management Information

1. Executive Information System (EIS)
2. TSCX
3. Data Warehouse
4. Intranet
5. Internet

Special Purpose

1. WINFORMS
    - Reports Distribution System (RDS)

2.1 Application Descriptions

Administrative applications at the Volpe Center include the following:

2.2 Finance Applications

2.2.1 AFIS Interface System (DIS)

Purpose: The purpose of the DIS system is to provide a front-end interface to the DOT DAFIS system and a back-end link from the DAFIS system to the Volpe Center FSOP systems.

Functionality: These two systems track all the spending – commitments, obligations, accrued expenditures, and disbursements – against most financial transactions at the Volpe Center. The Labor System and Acquisition Overhead do not interact with DIS.

Environment: Language / Data Source / Computer /System 1032, COBOL/ System 1032 Datasets, Flat files/ Compaq GS-140

2.2.2 Financial Status of Programs (FSOP)

Purpose: The FSOP (Financial Status of Programs) System is a part of the FSOP Accounting and Budget System in support of the Volpe Center's Code-80.

Functionality: The FSOP System Module encompasses the periodic reporting of funding status to Volpe Center Managers and to Sponsors. The data upon which the reports are based is maintained in an integrated FSOP database.

Environment: Language / Data Source / Computer /System 1032, COBOL/ S1032 Datasets, Flat Files/ Compaq GS-140

2.2.3 Funds Certification

Purpose The purpose of the Fund Certification System is to provide the Accounting Department the means to certify transactions and process the certified transactions.

Functionality: The Fund Certification System provides the Accounting Department the means to certify transactions, process the certified transactions, produce on demand reports to monitor the status of WPBs and PPAs, and to obtain on-line inquiries at the WPB and PPA levels which include printing capabilities. The Fund Certification System is a part of the FSOP Accounting and Budget System.

Environment: Language / Data Source / Computer/ System 1032, COBOL/S1032 Datasets, Flat Files / Compaq GS-140

2.2.4 Plan

Purpose The FSOP Plan System is a part of the FSOP Accounting and Budget System in support of the Volpe Center's Code-80 and assists these users in their daily activities.

Functionality The Plan System assists Budget users in maintaining the Project Plan Agreements (PPAs), the General Working Agreements (GWAs) and Reimbursable Agreements. The FSOP Plan System is a part of the FSOP Accounting and Budget System in support of Volpe Center's Code-80.

Environment: Language / Data Source / Computer/ System 1032, COBOL / S1032 Datasets/ Compaq GS-140

2.2.5 Total Cost

Purpose: The purpose of the Total Cost System is to accumulate commitments, obligations, accruals and disbursements from COAD and Labor by WPB cost element and PPA appropriation.

Functionality: The Total Cost System is a part of the FSOP Accounting and Budget System in support of the Volpe Center's Code-80. In addition to data accumulation, the Total Cost System produces the Direct Cost Summary, the Process Cost Summary, the Overhead Cost Summary, and the Labor Cost Summary Reports.

Environment: Language / Data Source / Computer/ System 1032, COBOL/ S1032 Datasets, Flat Files/ Compaq GS-140

2.2.6 Work Plan Budget (WPB)

Purpose: The purpose of the WPB system is to provide current WPB information. The WPB System is a part of the FSOP Accounting and Budget System.

Functionality: The WPB system enables Volpe Center Budget and Accounting Analysts to maintain all WPB records for the current fiscal year that are stored in the FSOP database, thereby providing current WPB information necessary to modify funding plans. The WPB System is a part of the FSOP Accounting and Budget System.

Environment: Language / Data Source / Computer/ System 1032, COBOL/ S1032 Datasets/ Compaq GS-140

2.2.7 Invoice Tracking

Purpose: The purpose of the Invoice Tracking system is to allow the Accounts Payable Department to track the status of vendor invoices and monitor payment activity.

Functionality: It consists of two subsystems: Invoice and Prompt Payment. The Invoice System is maintained with the Accounts Payable Department. Vendor invoice and prompt payment data is entered by a data entry clerk.

Environment: Language / Data Source / Computer/ ACCESS / ACCESS / Network Intel Server

2.2.8 Labor

Purpose: The purpose of the Labor Distribution System is to capture distribution of federal labor to projects and to provide management with a statement of federal employee labor charged to projects at the Volpe Center.

Functionality: The bi-weekly Labor Distribution System is divided into four production phases. Phase II consists of a final edit, the update of the LABOR ASCII Master file, the creation of a labor database, and the generation of reports. Phase III is run when the month end coincides with the end of pay cycle. Month end reports are generated in this cycle. Phase IV is run after Phase II when the month ends in

the midst of a pay cycle. In addition to generating monthly reports, the Labor Master File is updated in Phase IV with accrued data and a corresponding database is created.

Environment: Language / Data Source / Computer /COBOL, System 1032, Scope / Flat files, S1032 Datasets / Compaq GS-140

2.2.9 Payroll

Purpose: The purpose of the Volpe Center's Payroll System is to provide labor time to the CUPS system in Oklahoma City.

Functionality: The Volpe Center's Payroll System is generated from labor time that is processed in Atlanta, and then forwarded to Oklahoma City. A payroll skeleton file is electronically transmitted to the Volpe Center every pay period. A database is created from the file, and various payroll reports are generated using the database.

Environment: Language / Data Source / Computer/ COBOL, System 1032 / Flat files, S1032 / Compaq GS-140

2.2.10 Delphi (future)

Purpose: The Delphi System is a Department of Transportation wide financial system. It will be implemented at Volpe sometime in mid fiscal year 2003 and will provide the Volpe Center with all finance capabilities currently available.

2.3 Acquisition Applications

2.3.1 Contract Administration System (CAS)

Purpose: The purpose of the Contract Administration system is to track information about procurement contracts and contractors.

Functionality: The system keeps track of contractor skill descriptions, contract and contractor profiles, rates (hourly direct and indirect), modifications to contracts, and the agency responsible for auditing of a contract and contractor billing. Various reports are produced on a regular basis or as required.

Environment: Language / Data Source / Computer/ COBOL, System 1032, Scope / RMS Keyed files, S1032 / Compaq GS-140

2.3.2 Procurement Information Management (PRIM)

Purpose: The purpose of the PRIM system is to maintain procurement/contract information for the Acquisition Division.

Functionality: PRIM is an on-line/batch system. Users access the on-line system through a. menu program. The on-line system is used to make inquiry and data entry activities. The batch system is used to generate reports and interact with other system. This system has daily, weekly, monthly and quarterly reporting activities. The FSOP system is accessed to make some of these reports and to update the funding information on PRIM. Contract specialists use PRIM to report the status of contract procurements from time or receipt to obligating.

Environment: Language / Data Source / Computer /COBOL, System 1032 / RMS Keyed files, S1032 / Compaq GS-140

2.3.3 PRISM (future)

Purpose: PRISM is a COTS application currently being implemented at Volpe. It will replace all functionality provided by PRIM and then some. Full implementation is scheduled for some time in fiscal year 2003.

2.3.4 Reconstructed PRISM System

Purpose: The purpose of the Reconstructed PRISM System is to provide procurement information to the Receiving and Property Systems for procurements that use Simplified Acquisition Procedures (SAP).

Functionality: The Reconstructed PRISM system is used to assure data transfer and quality from the PRISM Acquisition system for the Receiving System and Property. This system has daily, weekly, monthly and quarterly reporting activities.

Environment: Language / Data Source / Computer/ SQL, COBOL, 1032, Scope / Flat files, S1032 / Compaq GS-140

2.4 Facilities

2.4.1 Property

Purpose: The purpose of the Property system is to provide data pertinent to personal property controlled by the Center.

Functionality: The Property Management System consists of the storage of all tagged personal property controlled by the Volpe Center. Data is entered via an on-line menu, that allows users to add, delete, change, transfer, etc., property. Weekly, Monthly, Quarterly and Yearly reports are generated via system generated batch processing.

Environment: Language / Data Source / Computer/ COBOL, System 1032, Scope / RMS Keyed files, S1032 / Compaq GS-140

2.4.2 Communication

Purpose: The Communications Management System was developed for the Finance and Administration Services Division of Volpe Center. The system provides an efficient and user friendly method for maintaining and reporting information relating to inventory, cost distribution and assignment of telephones within the Volpe Center complex.

Functionality: The function is accomplished with a monthly production cycle supported with appropriate ad-hoc data entry management tasks.

Environment: Language / Data Source / Computer/ COBOL, System 1032, Scope / Flat files, S1032 / Compaq GS-140

2.4.3 Supply

Purpose The purpose of the Supply System is to track inventory items that do not require property numbers.

Functionality The Supply System tracks stock issues, produces the Supply Catalog, and supports use of the Federal Supply Inventory. Stock issue data is provided to the DAFIS Interface System (DIS). Monthly reporting is also provided.

Environment: Language / Data Source / Computer/ COBOL, System 1032, Scope / S1032 / Compaq GS-140

2.5 Human Resources

2.5.1 Personnel

Purpose: The purpose of the Personnel System is to provide information concerning personnel at the Center and to provide a data source for other Volpe Center applications.

Functionality: The Personnel System is maintained by the FAA in Oklahoma City. Personnel Master file data is extracted from Oklahoma City's file monthly to populate the Center's Personnel Master file. This data is used to generate the Personnel Summary reports for Center management

Environment: Language / Data Source / Computer/ COBOL, System 1032, Scope / Flat files, S1032 / Compaq GS-140

2.5.2 Full Time Equivalency (FTE)

Purpose The purpose of the Full Time Equivalency System is to provide FTE information concerning personnel at the Center to Center management.

Functionality The Full Time Equivalency system is maintained by the FAA in Oklahoma City. FTE data is extracted monthly and used to update a fiscal year-to-date database. This data is used to generate the FTE reports for Center management

Environment: Language / Data Source / Computer/ COBOL, System 1032, Scope / Flat files, S1032 / Compaq GS-140

2.5.3 Minority

Purpose: The purpose of the Minority System is to provide the Center's Equal Employment Opportunity Officer with fiscal year-to-date promotions, attritions and accessions and to identify employees by ethnic background, grade, sex, and occupational code for use in a variety of reports.

Functionality: Data for the Minority system is maintained by the FAA in Oklahoma City in the CPMIS system. Data is extracted monthly and used to populate the Minority database at the Center.

Environment: Language / Data Source / Computer/ COBOL, System 1032, Scope / Flat files, S1032 / Compaq GS-140

2.6 Management Information

2.6.1 Executive Information System (EIS)

Purpose: The purpose of the EIS system is to provide access to information contained in various Volpe Center financial and administrative management systems.

Functionality: The EIS system enables Volpe Center users to develop projections and comparative analyses for their projects. It also provides query capability for current and periodic financial data for Volpe Center projects.

Environment: Language / Data Source / Computer /System 1032, SQL, Scope / Flat files, Oracle / Compaq GS-140

26.2 TSCX

Purpose: The purpose of the TSCX system is to provide access to information contained in various Volpe Center financial and administrative management systems. This system enables users to develop projections and comparative analyses.

Functionality: The TSCX system uses special programs for the systems it access to provide data to users. The system enables users to query data and print it out in hard copy. TSCX functionality is similar to EIS (above) functionality.

Environment: Language / Data Source / Computer/ COBOL, System 1032, Scope / RMS Keyed files, S1032 / Compaq GS-140

2.6.3 Data Warehouse (future)

Purpose: The purpose of the Data Warehouse is to provide access to information contained in various Volpe Center financial and administrative management systems. The Data Warehouse enables users to analyze this information by running standard or "canned" reports, creating their own new or "ad hoc" reports and queries, drilling-down to additional detail, exporting data to other formats, and presenting data in multiple formats, including graphs. User queries can be saved, shared with other end-users, and the resulting reports can be printed.

Functionality: The Data Warehouse consists of several components, including: source systems, a data staging area, data marts, and web-based business intelligence software. It currently provides access to project-related financial, travel, and labor information obtained from various operational systems

Environment: Language /Data Source Oracle /Computer/ Unix Shell Scripts/ORACLE/ Compaq GS-140, Sun Enterprise 450

2.6.4 Intranet

Purpose: The purpose of the Volpe Center's intranet (VolpeNet) is to support the exchange of internal information, internal information dissemination, and to support web-enabled solutions for business process streamlining where applicable.

Functionality: Via traditional static HTML code, as well as an Oracle database-driven dynamic code, web pages are served up and viewed through a web browser. In some cases information is restricted through an Oracle database log in.

AdNet, which used to be a sister site to VolpeNet focused on Acquisition information, is in the process of being incorporated into the overall intranet site and will no longer be a separate entity.

Environment: Language / Data Source/Computer/ java script, Perl, HTML, PL/SQL via oracle web products/ Volpe Data Staging Area/ Dell PowerEdge 6300 with NT

2.6.5 Internet

Purpose: The purpose of the Volpe Center's internet is to increase visibility, communicate with Volpe constituents, support outreach efforts, and share information with the public.

Functionality: Via traditional static HTML code, as well as an Oracle database-driven dynamic code, web pages are served up and viewed through a web browser.

Environment: Language/Data Source/Computer/java script, Perl, HTML, PL/SQL via ORACLE web products/Volpe Data Staging Area/ SUNE450 with Solaris UNIX

2.7 Special Purpose

2.7.1 WINFORMS

Purpose: The purpose of the WinForms System is to provide an electronic forms processing capability that automates the task of filling out and printing high volume forms used by the Volpe Center.

Functionality: Forms can be filled in and printed.

Environment: Language / Data Source / Computer/ FORMFLOW, Visual Basic, Visual C++, SQL/ Oracle, Clipper, dBase / Network Intel Server

2.7.2 Reports Distribution System (RDS)

Purpose: The purpose of the Reports Distribution system is to provide a mechanism for printing reports for specific organizations and/or individuals at the Center.

Functionality: The RDS system is an on-line system that contains data pertaining to reports distribution: who receives the report, what division they are in, how many copies to print. This system is internal to Administrative Applications.

Environment: Language / Data Source / Computer/ System 1032, Scope / S1032 / Compaq GS-140

OVERVIEW OF VOLPE CENTER NETWORK AND DESKTOP ENVIRONMENT

The Volpe Center network extends over seven buildings and provides access to local and remote computing resources for the Center's employees and contractors. The main building is a 12-story tower with all other buildings either physically connected to the tower or within a city block away (i.e., 5 Cambridge Place). The layout of the Center provides a right of way between six of the seven Center buildings. There is currently no conduit access to the 5 Cambridge Place facility (Computer Science Corporation (CSC) Contractor location) from the Volpe Center. Network connectivity between the Volpe Center and the CSC location is provided by four T1 circuits leased from Verizon Communications. The Volpe Center Infrastructure LAN/WAN Network diagram is available as Section J. 7 - LAN/WAN DIAGRAM.

The network is 100 percent built with Cisco 6000 and 5000 series Ethernet switches located at the central switching hub and the various communications closets. A brief description of the various network and desktop components is as follows:

Cabling

The Volpe Center has in place a fiber optic backbone network that interconnects all switches located in the various communications closets with the two central switches located in the Building 3 Data Center. The fiber is terminated by ST connectors within a properly enclosed patch panel in each communications closet and consists of a 6-pair, 62.5 x 125 micron cable, the only exception being the cable that connects the Building 3 Data Center with the second floor communications closet in Building 1. That cable is a 36-pair, 62.5 x 125 micron cable, the only two-to-one traffic concentration point to an otherwise all point-to-point fiber-connected network. The connections from the communications closets to the Volpe Center's offices and labs are via dual Category 5 copper wire drops. The 5 Cambridge Place location supporting CSC is connected to the Volpe network via four T1 circuits.

Network Equipment

The switching equipment used in the Volpe Center network is manufactured by Cisco Systems. There are over 25 Ethernet switches installed in communications closets located strategically throughout the Volpe Center. The majority of this hardware is Cisco 6000 Series which is capable of supporting 100 Mbps to the desktop with Gigabit connections between the closet switches and the main distribution switches in the second floor closet in Building 1 and in the Building 3 Data Center.

Network Management Systems

All Cisco Ethernet switches at the Volpe Center, including those at 5 Cambridge Place, are managed by two network monitoring systems, the Cisco Works 2000 and the Ipswitch WhatupGold, both on Windows NT 4.0. The Cisco Works 2000 is an interactive management system designed to integrate tightly into the Cisco SNMP agents on the Cisco routers and switches. This is managed from the Building 3 Network Engineering Control Center. The WhatupGold is also an interactive management system that

provides alarms to Network Engineering in the event of a failure of any of the network's routers, switches or servers.

5 Cambridge Place

Network access to 5 Cambridge Place from the Volpe Center is provided by 4 T1s. The T1s connect to Cisco 2514 routers that support the Decnet, IP, and IPX protocols. Cisco 5500 series Ethernet switches provide user connectivity via closet copper wiring that supports mostly 10Mbps to the desktop.

Servers

The Volpe Center has over 100 file servers in use by Government employees and Contractor employees. A combination of Unix, NT, and Novell servers are located at various locations throughout the Volpe Center. Approximately half of the servers are located in Building 3's Data Center and at the 5 Cambridge Place communications rooms. Other servers are located in or near the departments they serve.

Desktop Computers and Software

The desktop computer population consists primarily of Intel Pentium II/III/IV machines of various speed and memory configurations with Microsoft Windows and Office software, the current standard being Windows 98 and Office 2000.

Printers

The Volpe Center has about 100 network printers of various models and ages, mostly made by Hewlett Packard.

Service Desk Management

Service Desk Management is provided through the Heat Helpdesk and the ZenWorks Desktop management products.

Network Protocols

The Volpe Center's network has a high concentration of Novell Netware servers running the IPX/SPX protocol. There is also a significant amount of the TCP/IP family of protocols on the network to reach Unix, NT, and DEC machines. Other protocols on the Volpe Network include Netbeui, LAT, and Appletalk.

Electronic Mail

Currently, the Volpe Center mail system is based on the Microsoft Exchange product. The users are spread across a total of five servers.

Internet

The Volpe Center's Internet Service Provider (ISP) is Genuity. Access to the Internet is provided through dual bound T1s from the Cambridge office of Verizon Communications, backed up by a single T1 from its Boston office. A Cisco 2500 router

automatically switches the "hot backup" Boston connection whenever Genuity no longer "sees" its main Cisco 4700 router that controls the dual Cambridge connection. Both servers are located at the Volpe Center but are controlled by Genuity. The Volpe Center's network connection to the Internet is through a Symantec Firewall. This firewall is a highly rule-structured, proxy-based/content-filtering type that currently supports five isolated segments:

ISP or Internet Segment
Intra-Internet Web Segment
FMCSA / SAFER Segment
WAN Segment
Volpe Center Segment

The firewall is backed up by an identical hot stand-by configuration in case the active configuration fails.

Federal Aviation Administration's ADTN 2000

Access to the ADTN 2000 network from the Volpe Center network is via a Cisco 4700 router that performs Network Access Translation (NAT) for ADTN 2000's private address structure. The NAT router interfaces into an ADTN 2000 Cisco 7200 router that has two Wide Area Network (WAN) connections into the ADNT 2000 network: Jamaica, NY, and Burlington, MA.

DOT's Intermodal Data Network (IDN)

Access to the IDN from the Volpe Center network is through a Cisco 4700 router and dual bound T1s to the IDN in Washington, DC.

Surface Transportation Board (STB)

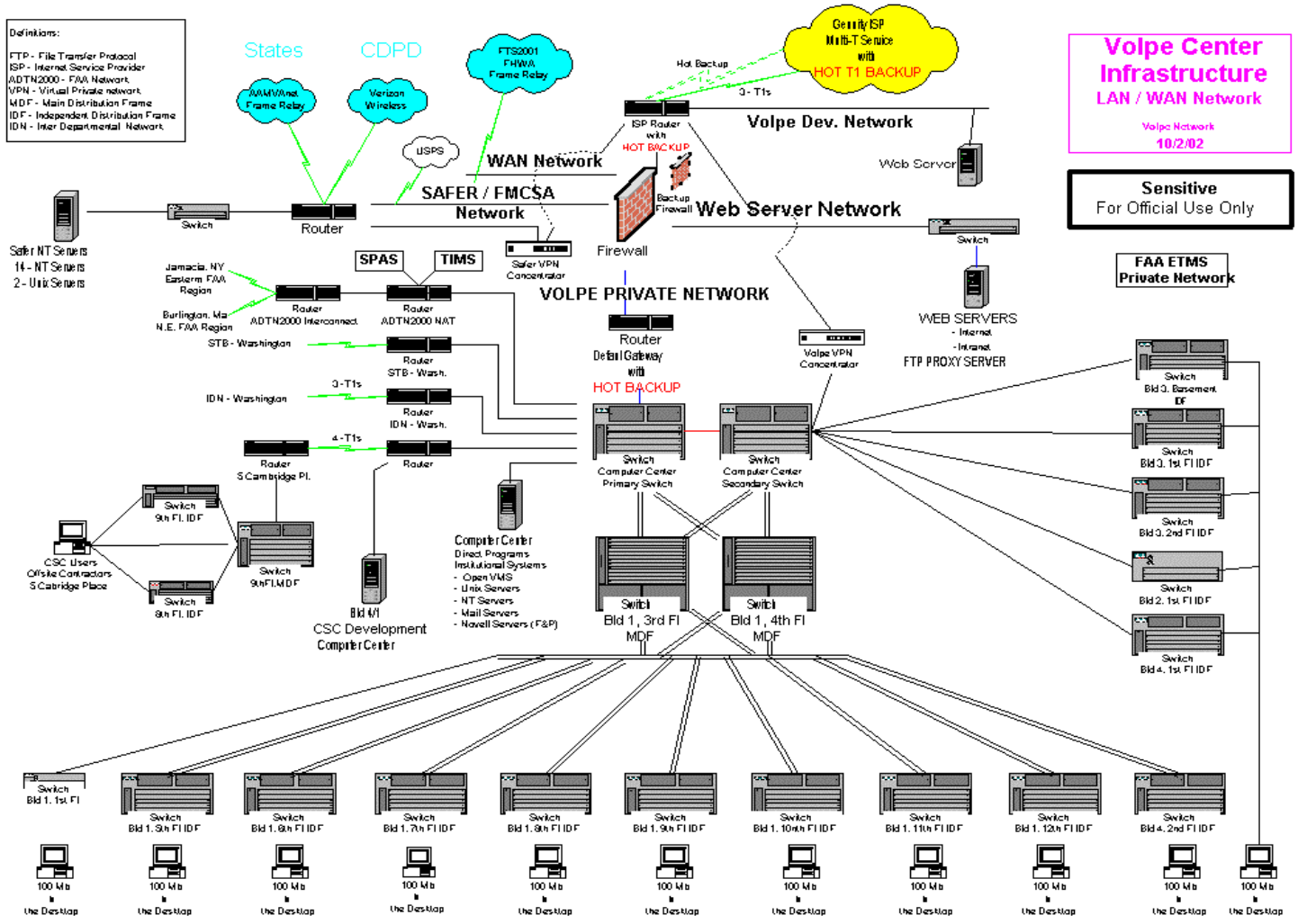
Access to the STB network from the Volpe Center network is through a Cisco 4xxx router that controls the single T1 connection between the two networks.

TIMS and SPAS

TIMS and SPAS represent separate networks within the Volpe Center. Cisco 4700 routers provide communications between the TIMS and SPAS networks and the Volpe Network.

Virtual Private Network (VPN)

During FY 2002, VPN access to all of the Volpe Center's networked resources will allow users to run telecommuting and remote desktop applications such as NetMeeting and Exchange WebMail.


Definitions:
FTP - File Transfer Protocol
ISP - Internet Service Provider
ADTN2000 - FAA Network
VPN - Virtual Private network
MDF - Main Distribution Frame
IDF - Independent Distribution Frame
IDN - Inter Departmental Network
States
CDPD
AAMVAnet Frame Relay
Verizon Wireless
FTS2001 FHWA Frame Relay
USPS
Genuity ISP Multi-T Service with HOT T1 BACKUP
Hot Backup
3 - T1s
Volpe Center Infrastructure
LAN / WAN Network
Volpe Network
10/2/02
Sensitive
For Official Use Only
Volpe Dev. Network
ISP Router with HOT BACKUP
WAN Network
Web Server
SAFER / FMCSA Network
Backup Firewall
Web Server Network
Safer NT Servers
14 - NT Servers
2 - Unix Servers
Switch
Router
Safer VPN Concentrator
Firewall
Switch
FAA ETMS Private Network
SPAS
TIMS
Jamacia, NY Eastern FAA Region
Burlington, Ma N.E. FAA Region
Router ADTN2000 Interconnect
Router ADTN2000 NAT
VOLPE PRIVATE NETWORK
WEB SERVERS
- Internet
- Intranet
FTP PROXY SERVER
STB - Washington
Router STB - Wash.
Router Default Gateway with HOT BACKUP
Volpe VPN Concentrator
IDN - Washington
3 - T1s
Router IDN - Wash.
4 - T1s
Router S Cambridge Pl.
Router
Switch Bld 3, Basement DF
Switch Bld 3, 1st FlIDF
Switch Bld 3, 2nd FlIDF
Switch Bld 2, 1st FlIDF
Switch Bld 4, 1st FlIDF
Switch Computer Center Primary Switch
Switch Computer Center Secondary Switch
Switch 9th Fl. IDF
Switch 8th Fl. IDF
Switch 9thFl.MDF
CSC Users Offsite Contractors 5 Cabridge Place
Bld 4/1 CSC Development Computer Center
Computer Center Direct Programs Institutional Systems
- Open VMS
- Unix Servers
- NT Servers
- Mail Servers
- Novell Servers (F&P)
Switch Bld 1, 3rd Fl MDF
Switch Bld 1, 4th Fl MDF
Switch Bld 1, 1st Fl
Switch Bld 1, 5th FlIDF
Switch Bld 1, 6th FlIDF
Switch Bld 1, 7th FlIDF
Switch Bld 1, 8th FlIDF
Switch Bld 1, 9th FlIDF
Switch Bld 1, 10th FlIDF
Switch Bld 1, 11th FlIDF
Switch Bld 1, 12th FlIDF
Switch Bld 4, 2nd FlIDF
100 Mb Ie Desktop

REVISION DATE: 8/16/02

Summary	
color	**platform**
NT Total	**75**
Project Use Supported	23
Institutional Use Supported	52
Windows 2000 Total	**12**
Project Use Supported	10
Institutional Use Supported	2
Novell Total	**14**
Project Use Supported	0
Institutional Use Supported	14
Unix Total	**23**
Project Use Supported	12
Institutional Use Supported	11
Net App Total	**2**
Project Use Supported	2
Institutional Use Supported	0
VMS Total	**8**
Project Use Supported	2
Institutional Use Supported	6
Linux RH 7.2 Total	**1**
Project Use Supported	0
Institutional Use Supported	1

Total Servers	**135**
Project Use Supported	49
Institutional Use Supported	86

Sensitive Information

Model	RAM	Processor	Speed	Number of Processors	Storage	Partition Info or Volume Info	Net	OS	Project Use Supported	Institutional Use Supported
Dell PowerEdge 2200	128	Intel Pentium II	233MHz	1				NT4		1
Dell PowerEdge 2200	128	Intel Pentium II	233MHz	1	Disk0=4G, Disk1=2G	Disk0=C 1G, X 3G, Disk1=O 2G	int	NT4		1
Gateway ALR 8200	128	Intel Pentium II	400MHz	1	Disk0=17G	Disk0=C 2G, D15G	svc	NT4		1
Dell PowerEdge 1400	512	Intel Pentium III	866MHz	2	2 8G RAID 0	Disk0=C 2G, D 1.5G, E 14G		NT4		1
								NT4		1
								NT4		1
								NT4		1
Thinkmate	128	Intel Pentium	300MHz	1	Disk0=6G	Disk0=C 3G, D 3G	int	NT4		1
Dell PowerEdge 2100/200	512	Intel Pentium II	200MHz	1	Disk0=2G, Disk1=8G	Disk0=C 1G, E 1G, Disk1=F 8.4G	int	NT4		1
Model American	256	Intel Pentium	200MHz	1	Disk0=4G	Disk0=C 502MB, D 3.5G	int	NT4		1
Dell PowerEdge 1400	256	Intel Pentuim III	866MHz	1	Disk0=8G	Disk0=C 2G, D 1G, E 5G	wan	NT4		1
								NT4		1
HP Pavillion 8562	64	Intel Pentium III	500MHz	1	Disk0=20G	Disk0=C 2G, D 17.5G	svc	NT4		1
No Name	128	Intel Pentium II	500MHz	1	Disk0=4G	Disk0= C 2G, D 320MB, E 1.7G	svc	NT4	1	
Compaq Proliant	512	Intel Pentium	133MHz	1	Disk0=1G, Disk1=2G, Disk2=1G, Disk3=1G	Disk0=C 1G, Disk1=F 2G, Disk2=E 1G, Disk3=G 1G	svc	NT4		1
Micron Millenia		Intel Pentium II								1
Dell PowerEdge 2300	256	Intel Pentium II	350MHz	2	Disk0=17G	Disk0=C 2G, 15G unused	int	NT4	1	
Dell PowerEdge 2300	128	Intel Pentium III	450MHz	2	Disk0=17G	Disk0=C 2G, D 500MB, E 14.5G	int	NT4	1	
Dell PowerEdge 2300	256	Intel Pentium II	450MHz	1	Disk0=8.5G	Disk0=C 2G, D 6.5G	int	NT4		1
Dell PowerEdge 2300	512	Intel Pentium III	450MHz	2	Disk0=8.5G, Disk1=8.5G	Disk0=C 2G, E 6.5G, Disk1=F 8.5G	svc	NT4	1	

Model	RAM	Processor	Speed	Number of Processors	Storage	Partition Info or Volume Info	Net	OS	Project Use Supported	Institutional Use Supported
Dell PowerEdge 1300	256	Intel Pentium III	500MHz	2	Disk0=8G, Disk1=8G	Disk0= C 2G, D 6G, Disk1=E 1G, F 7G	svc	NT4		1
Dell PowerEdge 1300	128	Intel Pentium III	500MHz	1	Disk0=9G (U3 SCSI 7200RPMs)	Disk0=C 2G, E 6G	wan	NT4	1	
Dell PowerEdge 2400	256	Intel Pentium III	700MHz	1	Disk0 = 17G	Disk 0=C 2G, D 15G	svc	NT4	1	
								NT4		1
Dell PowerEdge 2400	512	Intel Pentium III		2				NT4		1
PC Warehouse	32	Intel Pentium	200MHz	1	Disk0=400MB, Disk1=2G, Disk2=2G	Disk0= D 400MB, Disk1= C 1.5G, F 500MB, Disk2=E 2G	int	NT4	1	
Digital Prioris XL6200	96	Intel Pentium Pro	200MHz	1	Disk0=2.2G, Disk1=2G	Disk0=C 40MB, E=2G, Disk1=D 1G, F 1G	int	NT4	1	
Micron Client Pro	128	Intel Pentium III	500MHz	1	Disk0=13G	Disk0=C 2G, D=11G	int	NT4	1	
Dell PowerEdge 2300	520	Intel Pentium II	400MHz	2	Disk0=4G, Disk1=4G, Disk2=36G, Disk3=36G	Disk0=C 4G, Disk1=D 1.5G, G 2.5G, Disk 2=E 36G, Disk3=F 36G	int	NT4		1
Dell Dimension XPS R450	128	Intel Pentium II	450MHz	1	Disk0=9.6G, Disk1-36G	Disk0=C 8.6G, E 1G, D 36G	int	NT4		1
Dell PowerEdge 2300	512	Intel Pentuim III	450MHz	2	Disk0=8.4G, Disk1=8.4, Disk2=8.4, Disk3=8.4	Disk0=C 1.95G, D 6.5G, Disk1=E 8.4G, Disk2=F NO FS, Disk3=G 8.4G	svc	NT4		1
Dell PowerEdge 2200	128	Intel Pentium II	266MHz	1	Disk0=4G	Disk0=C 2G, E 2G	int	NT4		1

Model	RAM	Processor	Speed	Number of Processors	Storage	Partition Info or Volume Info	Net	OS	Project Use Supported	Institutional Use Supported
Dell PowerEdge 6300	1G	Intel Pentium Xeon	400MHz	2	Disk0=17G (RAID5 3 8.2G drives), Disk1=36G, Disk2=36G	Disk0=C 4G, D 3G, G 9G, Disk1=E 36G, Disk2=F 36G	int	NT4		1
Dell PowerEdge 1400SC	256	Intel Pentium III	933MHz	1	Disk0=16G	Disk0=C 2G, D 640MB, E 14G	int	NT4		1
Dell Precision WS 610MT	768	Intel Pentium II XEON		1	Disk0=34G	Disk0=C 4G, D 2G, E 28G	int	NT4		1
Dell PowerEdge 1300	128	Intel Pentuim III	500MHz	1	Disk0=8G	Disk0=C 7.5G, 500MB Unused	svc	NT4		1
Micron ClientPro		Intel Pentium III		1			svc	NT4	1	
								NT4	1	
Dell OptiPlex GX1	128	Intel Pentium III	500MHz	1	Disk0=9G	Disk0=C 2G, F 7G	svc	NT4		1
Digital Mod # 892WW	256	Intel Pentium	100MHz	4	Disk0=1G, Disk1=1G, Disk2=4G	Disk0=C 1G, Disk1=D 1G, Disk2=E4G	wan	NT4	1	
Dell PowerEdge 400	128	Intel Pentium III	800MHz	1	Disk0=20G	Disk0=C 2G, D 500MB, E 17.5G	wan	NT4	1	
Compaq Proliant 6000	1G	Intel Pentium II XEON	450MHz	4	Disk0=4G, Disk1=8G, Disk2=12G	Disk0=C 4G, Disk1=D 8G, Disk2=E 12G	wan	NT4	1	
								NT4	1	
Compaq Proliant 6000	1G	Intel Pentuim II XEON	500MHz	4	Disk0=43G, Disk1=8G	Disk0=C 4G, D 10G, E 29G, Disk1=F 8G	wan	NT4	1	
Compaq Proliant 6000	1G	Intel Pentium II XEON	500MHz	4	Disk0=42G	Disk0=C 3.94G, D 9.99G, E 28.3G	wan	NT4	1	
Compaq Proliant ML570	1.5G	Intel Pentium II XEON	700MHz	4	Disk0=17G, Disk1=69G	Disk0=F 17G, Disk1=C 3.94G, D 17.5G, E 46.26G	wan	NT4	1	

Model	RAM	Processor	Speed	Number of Processors	Storage	Partition Info or Volume Info	Net	OS	Project Use Supported	Institutional Use Supported
	128	Intel Pentium	200MHz	1	Disk0=2G, Disk1=2G	Disk0=C 2G, Disk1=D 2G	wan	NT4	1	
Micron Millennia LXE	64	Intel Pentium	200MHz	1	Disk0=4G	Disk0=C 2G	wan	NT4	1	
Dell PowerEdge 2200	256	Intel Pentium II	233MHz	1	Disk0=4G, Disk1=4G	Disk0=C 2G, E 2G, Disk1=D 1G, F 3G	wan	NT4		1
Dell PowerEdge 1300	256	Intel Pentium III	500MHz	1	Disk0=8.6G	Disk0=C 8G	svc	NT4		1
Dell PowerEdge 1300	256	Intel Pentium III	500MHz	1	Disk0=8.2G	Disk0=C 2G, D 750MB, E 6G	svc	NT4	1	
Compaq Proliant ML350	196	Intel Pentium III	200MHz	1	Disk0=2.9G, Disk1=14G	Disk0=C 2.9G, Disk1=D 14G	svc	NT4	1	
Micron Millennia	128	Intel Pentium II	400MHz	1	Disk0=8G	Disk0=C 2G, D 6G	int	NT4		1
Compaq Proliant ML370	384	Intel Pentium III	1GHz	1	Disk0=18G	Disk0=C 2G, E 15G	wan	NT4		1
Dell PowerEdge 1400sc	256MB	Intel Pentium III	933Mhz	1	Disk0=17G	Disk0=C 3G, D 768MB, E 14G	wan	NT4		1
Dell PowerEdge1300	264	Intel Pentium II	400MHz	1	Disk0=8.5G, Disk1=8.5G	Disk0=C 2G, 6.5GUnpartitioned, Disk1=D 8.5G	int	NT4		1
Dell PowerEdge 1300	264	Intel Pentium II	500MHz	2	Disk0=8.5G, Disk1=8.5G	Disk0=C 2G, D 6.5G, Disk1=E 1G(pg)	int	NT4		1
Dell PowerEdge 1300	264	Intel Pentium II	500MHz	2	Disk0=8.5G	Disk0=C 2G, D 6.4G	svc	NT4		1
Dell PowerEdge 1300	264	Intel Pentium II	500MHz	1	Disk0=8.5G	Disk0=C 2G, D 6.4G	wan	NT4		1
Compaq Proliant 2500	256	Intel Pentium Pro	200MHz	2	4 4.3G drives		int	NT4		1
Dell PowerEdge 1300	256	Intel Pentium II						NT4		1
Dell PowerEdge 2300	264	Intel Pentium II	350MHz	1	Disk0=12.6G	Disk0= C 2G, D 10.3G	int	NT4		1
Dell PowerEdge 4400	1G	Intel Pentium III XEON	900MHz	2			int	NT4		1
Dell PowerEdge 4400	1G	Intel Pentium III XEON	900MHz	2			int	NT4		1
Dell PowerEdge 4400	1G	Intel Pentium III XEON	900MHz	2			int	NT4		1

Model	RAM	Processor	Speed	Number of Processors	Storage	Partition Info or Volume Info	Net	OS	Project Use Supported	Institutional Use Supported
Dell PowerEdge 4400	1G	Intel Pentium III XEON	900MHz	2			int	NT4		1
Dell PowerEdge 1300	128	Intel Pentium II	400MHz	1	Disk0=4G, Disk1=4G RAID1	Disk0=C 2G, D 2G	int	NT4		1
Micron Millennia	80	Intel Pentium	166MHz	1	Disk0=1.2G, Disk1=520MB	Disk0=C 1.2G, Disk1=D 520MB (swp)	int	NT4		1
Dell Dimension XPS P90c	64	Intel Pentium	90MHz	1	Disk0=2G	Disk0=C 2G	int	NT4		1
Digital Alpha Station 200 4/166	64	DEC-321064		1	Disk0=1G	Disk0=C 16MB, D 985MB	int	NT4		1
Dell PowerEdge 1300	128	Intel Pentium II	400 MHz	1	Disk0=4G, Disk1=4G RAID1	Disk0=C 2G, D 2G	int	NT4		1
Dell PowerEdge 1300	128	Intel Pentium II	401 MHz	2	Disk0=4G, Disk1=4G RAID2	Disk0=C 2G, D 2G	int	NT4		1
Dell PowerEdge 4200	256	Intel Pentium II	233MHz	2	Disk0=8G	Disk0=C 2G, E 6G	int	NT4	1	
Dell PowerEdge 2300	256	Intel Pentium II	400MHz	1	Disk0=9G, Disk1=9G	Disk0=C 2G, D 6.45G, Disk1=Free Space	int	NT4		1
Dell PowerEdge 300	128	Intel Pentium III	800MHz	1	Disk0=19G	Disk0=C 2G, D 16G, E 400MB	int	NT4		1

Model	RAM	Processor	Speed	Number of Processors	Storage	Partition Info or Volume Info	Net	OS	Project Use Supported	Institutional Use Supported
Compaq Proliant ML370	512	Intel Pentium III	933MHz	1	Disk0=52G (RAID 0)	Disk0=C 2G, D 1.5G, E 48G	wan	Win2K		1
Dell PowerEdge 6400	1G	Intel Pentium III XEON	1GHz	2	Disk0(RAID5 4X36G)=108G	Disk0=C 5G, D 2.5G, E 94G	wan	Win2K		1
Dell PowerEdge 2400	IG	Intel Pentium III	600MHz	2			int	Win2K	1	
Dell PowerEdge 2400	1G	Intel Pentium III	1GHz	2	RAID 5 4 drives, 1 logical drive, 1 hot spare	Disk0=C 5G, D 2.5G, E 40G	int	Win2K	1	
Dell PowerEdge 2400	1G	Intel Pentium III	600MHz	2			int	Win2K	1	
Compaq Proliant ML370	2G	Intel Pentium III	1.3GHz	2	RAID 5 5 12.8G drives, 1 logical drive	Disk0=C 4G, D 5G, E	int	Win2K	1	
Dell PowerEdge 2400	1G	Intel Pentium III	1GHz	2	RAID 5 4 drives, 1 logical drive, 1 hot spare	Disk0=C 5G, D 2.5G, E 40G	int	Win2K	1	
Dell PowerEdge 2300	256	Intel Pentium III		1	Disk0=8G	Disk 0=C 8G	svc	Win2K	1	
Dell PowerEdge 1500SC	1G	Intel Pentium III		1		Disk0=C 5G, Disk1=D 2.5G, Disk2=E 9G, Disk3=G 17G, Disk4=H 17G, Disk5=I 17G, Disk6=J 17G	int	Win2K	1	
								Win2K	1	
								Win2K	1	
								Win2000 Professional	1	
Micron ClientPro Vxe	130 MB	Intel P54/P55	66 Mhz	1	1 4 GB drive	SYS 4GB	int	Netware 4.11 / 2 user		1
Dell Poweredge 2300	65 MB	Intel Pentium III	350 Mhz	1	3, 9 GB drives using RAID 5	SYS 5GB, VOL1 12 GB	int	Netware 4.11 / 100 user		1
Dell Poweredge 1300	65 MB	Intel Pentium III	350 Mhz	1	1 8 GB	SYS 8GB	int	Netware 4.2 / 15 user		1
Dell Poweredge 2400	524 MB	Intel Pentium III	450 Mhz	1	4, 9 GB drives using RAID 5	SYS 2GB, VOL1 3GB, VOL2 8GB, VOL_PST 12GB	int	Netware 4.11 / 100 user		1

Model	RAM	Processor	Speed	Number of Processors	Storage	Partition Info or Volume Info	Net	OS	Project Use Supported	Institutional Use Supported
					4, 9 GB drives	SYS 2GB, VOL1 2GB, VOL2 3GB, VOL3 2GB, VOL4 2 GB, VOL_PST 14 GB	int			1
Dell Poweredge 2400	524 MB	Intel Pentium III	450 Mhz	1	using RAID 5			Netware 4.11 / 100 user		
					4 Drives, 3 8GB	SYS 4GB, VOL1 7 GB, VOL2 8 GB, VOL5 1.6GB, VOL_PST 8 GB	int			1
Dell Poweredge 1300	392 MB	Intel Pentium III	350 Mhz	1	and 1 4GB			Netware 5.0 / 250 user		
					6 Drives, 2 2GB, 4GB, 18GB, 2	SYS 2 GB, VOL1 1GB, VOL2 1GB, VOL3 8GB, VOL4 9GB, VOL5 4GB, VOL6 8GB, VOL_PST 8.6GB	int			1
Dell Poweredge 2100	426 MB	Intel Pentium III	200 Mhz	1	8GB			Netware 4.11/ 150 user		
					6, 8GB drives	SYS 4GB, VOL1 4GB, VOL2 8GB, VOL3 8GB, VOL4 8GB, VOL_PST 8GB, VOL_PST2 8GB	int			1
Dell Poweredge 2400	524 MB	Intel Pentium III	450 Mhz	1				Netware 5.0 / 150 user		
					4, 8 GB drives. 3 of them are in a	SYS 12GB, VOL1 1GB, VOL2 4GB, VOL_PST 8GB	int			1
Dell Poweredge 2300	524 MB	Intel Pentium III	350 Mhz	1	RAID 5 setup			Netware 5.0 / 250 user		

Model	RAM	Processor	Speed	Number of Processors	Storage	Partition Info or Volume Info	Net	OS	Project Use Supported	Institutional Use Supported
						SYS 2GB, VOL1 6GB, VOL2 4GB, VOL3 4GB, GRAPHICS 16GB, VOL_PST 8GB	int	Netware 4.11 / 125 user		1
Dell Poweredge 4200	262 MB	Intel Pentium II	333 Mhz	1	5, 8 GB drives					
Generic 233 MHZ	130 MB	Intel Pentium	233 Mhz	1	1, 4 GB drive	SYS 3 GB	int	Netware 4.10 / 250 user		1
						SYS 3GB, VOL1 3GB, VOL2 3GB, VOL3 4GB, SUPPORT 8GB, VOL_PST 8GB	int	Netware 5.0 / 100 user		1
Dell Poweredge 2300	524 MB	Intel Pentium III	450 Mhz	1	5, 8 GB drives					
Dell Poweredge XE590	195 MB	Intel i486DX2	66 Mhz	1	4, 4GB RAID 5 Array	SYS 9 GB, VOL1 2GB	int	Netware 4.10 / 500 user		1
Dell Poweredge 2300	524 MB	Intel Pentium III	350 Mhz	1	3, 4 GB dives in RAID 5 Array	SYS 8GB	int	Netware 4.11 / 250 user		1
					2 gig HD, 500 MB HD = **2.5**					1
Dec AlphaServer 400	256	Alpha AXP	233 MHz	1	**gig total**		svc	Digital UNIX 4.0F		
SGI Octane	1G	SGI R10000	175 MHz	1	4G		int	IRIX 6.5.6	1	
Sunfire 280 R	2G	UltraSparc III	400MHz	2	72GB		INT	Solaris 8		1
Sun 280R								Solaris 8	1	
Sun 280R								Solaris 8	1	
									1	
Sun SPARCstation 5	64	Sun SPARC	110MHz	1	1 gig HD, 4 gig HD = **5 gig total**		svc	Solaris 7		
SGI Indigo2	256	SGI R4000	250 MHz	1	9G		int	IRIX 6.5.9	1	
										1
Sun Enterprise E450	1024	UltraSPARC II	400 Mhz	2	2x9GB RAID1, 5x18GB RAID5 w/hot spare		svc	Solaris 7		
Sun Enterprise E450	2GB	Sun UltraSparc II	480MHz	2	72GB		svc	Solaris 7		1
Sun SPARCstation 5	64	Sun SPARC	110MHz	1	2.1 gig HD		svc	Solaris 7		1
AlphaStation 333	128	Alpha AXP	333 MHz	1	4 gig total HD			Digital UNIX 4.0F		1

Model	RAM	Processor	Speed	Number of Processors	Storage	Partition Info or Volume Info	Net	OS	Project Use Supported	Institutional Use Supported
					4 gig HD, 9gig HD = **13 gig**				1	
SGI Octane	704	SGI R10000	175 MHz	2	**Total**		int	IRIX 6.5.9		
AlphaStation 333	128	Alpha AXP	333 MHz	1				Digital UNIX 4.0F		1
SGI Octane	512	SGI R10000	195 MHz	1	13G		int	IRIX 6.4		1
										1
Sunfire 280 R	1 gig	UltraSPARC III	1 G	1	40G		Safer	Solaris 8		
					4.2 gig HD, 4.2 gig HD = **8.4 gig**					1
Sun UltraSPARC 1	320	Sun UltraSPARC	167MHz	1	**total**		svc	Solaris 7		
SGI Octane	512	SGI R10000	250 MHz	1	24G		int	IRIX 6.5.9	1	
SGI Octane	512	SGI R10000	175 MHz	1	13G		int	IRIX 6.5	1	
AlphaServer ES40	2G	Alpha AXP	833 MHz	2			int	Tru64 Unix V5.1	1	
AlphaServer 4000 5/300	1G	EV5 Alpha	300 MHz	1	25.5 GB		int	Tru64 Unix V5.0		1
SGI Octane	512	SGI R10000	175 MHz	1	4G		int	IRIX 6.5.6	1	
					1 gig HD, 1 gig HD, 2 gig HD =				1	
DEC AlphaServer 400	32	Alpha AXP	233 MHz	1	**4 gig total**		svc	Digital UNIX 4.0D		
SGI Origin 2000	2G	SGI R10000	195 MHz	4	54G		int	IRIX 6.5.6	1	
					14 18GB FC-AL drives = **252 GB**				1	
Net App F720		Alpha AXP			**total**		int	ONTAP 5.2.3		
					14 18GB FC-AL drives = **252 GB**				1	
Net App F720		Alpha AXP			**total**		int	ONTAP 5.2.3		
Dell								Linux Redhat 7.2		1
								OpenVMS AXP	1	
AlphaServer 4000 5/400	512	EV5 Alpha	400 MHz	1	21.5 GB		int	V7.1-2		
								OpenVMS VAX		1
VAXstation 4000-60	40	VAX	60 MHz	1	4 GB		int	V5.5-2		
								OpenVMS VAX		1
VAXstation 4000-60	40	VAX	60 MHz	1	1.7 GB		int	V7.1		
								OpenVMS AXP		1
AlphaServer GS-140	4 G	EV6 Alpha	525 MHz	2	4 GB		int	V7.1-2		
								OpenVMS AXP	1	
AlphaServer 7000 6/200	384	EV5 Alpha	200 MHz	2	N/A		int	V7.2-1		
								OpenVMS AXP		1
AlphaServer GS-140	4.5 G	EV6 Alpha	525 MHz	2	242 GB		int	V7.2-1		
								OpenVMS VAX		1
VAXstation 4000-90	64	VAX	90 MHz	1	2.3 GB		int	V5.5-2		
								OpenVMS VAX		1
VAXstation 4000-60	40	VAX	60 MHz	1	2 GB		int	V5.5-2		

FIPS PUB 106

GUIDELINE

ON

SOFTWARE MAINTENANCE

FIPS PUB 106

FEDERAL INFORMATION

PROCESSING STANDARDS PUBLICATION

CATEGORY: SOFTWARE

SUBCATEGORY: SOFTWARE MAINTENANCE

U.S. DEPARTMENT OF COMMERCE, Malcolm Baldrige,
Secretary

National Bureau of Standards, Ernest Ambler,
Director

Foreword

The Federal Information Processing Standards Publication Series of the National Bureau of Standards is the official publication relating to standards adopted and promulgated under the provision of Public Law 89-306 (Brooks Act) and under Part 6 of Title 15, Code of Federal Regulations. These legislative and executive mandates have given the Secretary of Commerce important responsibilities for improving the utilization and management of computers and automatic data processing in the Federal Government. To carry out the Secretary's responsibilities, the NBS, through its Institute for Computer Sciences and Technology, provides leadership, technical guidance, and coordination of government efforts in the development of guidelines and standards in these areas.

Comments concerning Federal Information Processing Standards Publications are welcomed and should be addressed to the Director, Institute for Computer Sciences and Technology, National Bureau of Standards, Gaithersburg, MD 20899.

James H. Burrows, Director
Institute for Computer Sciences and Technology

Abstract

There is a need for a strong, disciplined, clearly-defined approach to software maintenance. This report emphasizes the

importance of the consideration of software maintenance throughout the lifecycle of a software system and stresses the need to plan, develop, use and maintain a software system with future software maintenance in mind. General and functional definitions of software maintenance are provided and software change activities are identified. The report presents guidance for controlling and improving the software maintenance process and includes suggested criteria for deciding whether continued maintenance of a software system is justified. It concludes that an organization's software maintenance efforts can be improved through the institution and enforcement of software maintenance policies, standards, procedures, and techniques.

Key words: adaptive maintenance; corrective maintenance; Federal Information Processing Standards Publications; management; perfective maintenance; software engineering; software lifecycle; software maintenance; software maintenance management; software maintenance tools.

Federal Information

Processing Standards Publication 106

1984 June 15

ANNOUNCING THE

GUIDELINE ON SOFTWARE MAINTENANCE

Federal Information Processing Standards Publications are issued by the National Bureau of Standards pursuant to the Federal Property and Administrative Services Act of 1949, amended, Public law 89-306 (79 Stat. 1127) and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 Code of Federal Regulations (CFR)

Name of Guideline: Guideline on Software Maintenance.

Category of Guideline: Software, Software Maintenance.

Explanation: This Guideline presents information on techniques, procedures, and methodologies to employ throughout the lifecycle of a software system to improve the maintainability of that system. Guidance is presented on controlling and improving software maintenance. Also included is a glossary of technical terms, a list of supporting ICST publications and a list of suggested additional reading. Appendices provide information on the software maintenance process; how to decide whether or not to continue maintaining a system; and software maintenance tools. This Guideline is intended for use by both managers and maintainers.

Approving Authority: U.S. Department of Commerce, National Bureau of Standards, Institute for Computer Sciences and Technology.

Maintenance Agency: U.S. Department of Commerce, National Bureau of Standards, Institute for Computer Sciences and Technology.

Cross Index: None

Applicability: This Guideline is intended as a basic reference guide for Federal ADP managers and software maintainers for maintaining software systems throughout their lifecycle. It addresses both management and technical issues. Use of this Guideline is encouraged, but not mandatory.

Implementation: This Guideline should be consulted whenever Federal departments or agencies are developing or maintaining software; developing policies and procedures for developing or maintaining software; or considering alternatives to continued maintenance of a software system.

Specifications: Federal Information Processing Standards Publication 106 (FIPS PUB 106), Guideline on Software Maintenance (affixed)

Qualifications: The techniques and procedures presented in this Guideline are recommended for use in all Federal ADP organizations Specific environments, organizational priorities, available budget and staff resources, and many other factors should be taken into account when implementing these recommendations. The resulting organizational standards and/or guidelines should reflect the specific needs of the organization.

Where To Obtain Copies: Copies of this publication are for sale by the National Technical Information Service, U.S. Department of Commerce, Springfield, VA 22161. When ordering, refer to Federal Information Processing Standards Publication 106 (FIPSPUB106), and title. When microfiche is desired, this should be specified. Payment may be made by check, money order, or NTIS deposit account.

1. INTRODUCTION

Software maintenance accounts for approximately 60% to 70% of the application software resources expended within the Federal Governments In addition, the rapidly growing inventory of software systems is increasing the demand for software maintenance. Improved productivity in maintaining software, however, can offset these increases. Thus, the issue which must be addressed is not reducing the absolute cost, but rather improving the quality and effectiveness of software maintenance.

This Guideline provides general guidance for managing software maintenance. It presents an overview of techniques and procedures designed to assist management in controlling and improving the software maintenance process. It addresses the need for a software maintenance policy with enforceable controls for use throughout the software lifecycle, the management of software maintainers, and management methods which can improve software maintenance. It concludes that improvements in the area of software maintenance will come primarily as a result of the software maintenance policies, standards, procedures, and

techniques instituted and enforced by management

This Guideline is intended for use by both managers and maintainers. It addresses the need for a strong, disciplined, clearly defined approach to software maintenance. It emphasizes that the maintainability of the software must be taken into consideration throughout the lifecycle of a software system. Software must be planned, developed, used, and maintained with future software maintenance in mind. The techniques and procedures which are discussed in this Guideline are recommended for use in all Federal ADP organizations. Specific environments, organizational priorities, available resources (budget and staff), and many other factors also must be taken into account, however. The resulting organizational standards and guidelines should reflect the specific need of the organization.

This Guideline is divided into five sections and two appendices:

Section 1-INTRODUCTION discusses the purpose of this Guideline.

Section 2-SOFTWARE MAINTENANCE DEFINITION presents general and functional definitions of software maintenance

Section 3-THE SOFTWARE MAINTENANCE PROCESS provides a correlation between the software lifecycle and a software maintenance lifecycle, and identifies software change activities.

Section 4-CONTROLLING AND IMPROVING SOFTWARE MAINTENANCE provides the actual guidance on techniques and procedures to aid both management and the maintainer.

Section 5-SOFTWARE MAINTENANCE VS SYSTEM REDESIGN discusses factors which should be considered when deciding whether to redevelop or to continue maintenance of a software system.

Appendix I lists related reports published by the Institute for Computer Sciences and Technology of the National Bureau of Standards. Appendix II provides references to selected readings on software maintenance. A Glossary of frequently used software maintenance terms is also provided.

2. SOFTWARE MAINTENANCE DEFINITION

Software maintenance is the performance of those activities required to keep a software system operational and responsive after it is accepted and placed into production.

Software maintenance is the set of activities which result in changes to the originally accepted (baseline) product set. These changes consist of corrections, insertions, deletions, extensions, and enhancements to the baseline system. Generally, these changes are made in order to keep the system functioning

in an evolving, expanding user and operational environment.

Functionally, software maintenance activities can be divided into three categories: perfective, adaptive, and corrective.

Perfective maintenance includes all changes, insertions, deletions, modifications, extensions, and enhancements which are made to a system to meet the evolving and/or expanding needs of the user. Perfective maintenance refers to enhancements made to improve software performance, maintainability, or understandability. It is generally performed as a result of new or changing requirements, or in an attempt to augment or fine tune the software Activities designed to make the code easier to understand and to work with, such as restructuring or documentation updates (often referred to as "preventive" maintenance) and optimization of code to make it run faster or use storage more efficient]y are also included in the perfective category. Perfective maintenance comprises approximately 60% of all software maintenance.

Adaptive maintenance consists of any effort which is initiated as a result of changes in the environment in which a software system must operate. These environmental changes are normally beyond the control of the software maintainer and consist primarily of changes to the:

- rules, laws, and regulations that affect the system
- hardware configurations e.g., new terminals, local printers
- data formats, file structures
- system software, e.g., operating systems, compilers, utilities.

Approximately 20% of software maintenance falls into the adaptive category.

Corrective maintenance refers to changes necessitated by actual errors (induced or residual "bugs") in a system. Corrective maintenance consists of activities normally considered to be error correction required to keep the system operational By its nature, corrective maintenance is usually a reactive process where an error must be fixed immediately. Not all corrective maintenance is performed in this immediate response mode; but all corrective maintenance is related to the system not performing as originally intended. Corrective maintenance accounts for approximately 20% of all software maintenance. The three main causes of corrective maintenance are: (1) design errors, (2) logic errors, and (3) coding errors.

3. THE SOFTWARE MAINTENANCE PROCESS

The lifecycle of computer software covers its existence from its conception until the time it is no longer available for use. There are a number of definitions of the software lifecycle

which differ primarily in the categorization of activities or phases. One traditional definition is: requirements, design, implementation, testing and operation and maintenance.

The requirements phase encompasses problem definition and analysis, statement of project objectives, preliminary system analysis, functional specification, and design constraints. The design phase includes the generation of software component definition, data definition, and interfaces which are then verified against the requirements. The implementation phase entails program code generation, unit tests, and documentation. During the test phase, system integration of software components and system acceptance tests are performed against the requirements. The operations and maintenance phase covers the use and maintenance of the system. The beginning of the maintenance phase of the lifecycle is usually at the delivery and user acceptance of the software product set.

One way of describing the activities of software maintenance is to identify them as successive iterations of the first four phases of the software lifecycle, i.e., requirements, design, implementation. and testing.
Software maintenance involves many of the same activities associated with software development but also has unique characteristics of its own.

1. Maintenance activities are performed within the context of an existing framework or system. The maintainer must make changes within the existing design and code structure constraints. The older the system, the more challenging and time.consuming the software maintenance effort often becomes.

2. A software maintenance effort is typically performed within a much shorter time frame than a development effort. A software development effort may span one, two, or more years while corrective maintenance may be required within hours and perfective maintenance in cycles of, 1 to 6 months.

3. Development efforts must create all of the test data from scratch. Maintenance efforts typically can take advantage of existing test data and perform regression tests. The major challenge for the maintainer is to create new data to adequately test the changes to the system and their impact on the rest of the system.

The process of implementing a change to a production system is complex and involves many people in addition to the maintainer. This process begins when the need for, a change arises and ends after the user has accepted the modified system and all documentation has been satisfactorily updated.

Although the software maintenance process is presented in a linear fashion in figure 1, there are a number of steps where iterative loops often occur. The change request may be returned to the user for additional clarification, the results of the design review may necessitate additional design analysis or even modification of the change request; testing may result in additional design changes or recoding; the standards audit may require changes to the design documents, code, and/or documentation; and the failure of the users to accept the system may result in return to a previous step or the cancellation of the task.

Not all of the steps presented here, however, must be performed For each change. There are several points at which the process may end. The key is to ensure that each person or group impacted by a change is involved in the process, aware of the actions taken, and satisfied with the results.

4. CONTROLLING AND IMPROVING SOFTWARE MAINTENANCE

Software maintenance must be performed in a structured, controlled manner. It is simply not enough to get a system "up and running" after it breaks. Proper management control must be exercised over the entire process. In addition to controlling the budget, schedule, and staff, it is essential that the software maintenance manager control the system and the changes to it.

A great deal of code is not developed with maintenance in mind. Indeed, the emphasis has often been to get the program up and running without being "hindered" by guidelines, methodologies, or other controls. In addition, over the lifecycle of a software system, code and logic which may have been well designed and implemented often deteriorate due to an endless succession of "quick fixes" and patches which are neither well designed nor well documented. A system must not only be developed "with maintenance in mind," it must also be "maintained with future maintenance in mind." If this is done, the quality and maintainability of the code actually can improve. Otherwise, today's maintainable systems are destined to become tomorrow's un-maintainable systems.

4.1 Controlling Software Maintenance

The quality and maintainability of a software system often decrease as a system grows older. This is the result of many factors which, taken individually, may not seem significant. They are, however, cumulative and often result in a system which is very difficult to maintain. Quality programming capabilities and techniques are readily available. Until a firm discipline is defined for the performance of software maintenance, and that discipline is enforced, many systems will be permitted to deteriorate to the point where they are impossible to maintain.

The goal of software maintenance management is to keep

systems functioning and to respond to user requests in a timely and satisfactory manner. Given the realities of staffing limitations, computer resource limitations, and the user request backlog, this goal is very difficult to achieve, The realistic goal, then, is to keep the software maintenance process orderly and under control. The specific responsibility of the software maintenance manager is to keep application systems running and to facilitate communication between management, users and maintainers.

Controlling software maintenance is primarily maintaining an orderly process in which all requests are formally submitted, reviewed, assigned a priority, and scheduled. This does not mean that unnecessary delays should be built into the process, or that in small organizations these steps are not consolidated. Rather, it defines a philosophical approach which can help the software maintenance manager bring order to the software maintenance environment.

It is very rare for even a "perfect" system not to require significant maintenance during its lifetime. While software does not "break" in the sense that a piece of software can fail, it can become non-functional, or faulty due to changes in the environment in which it must operate, the size or sophistication of the user community, the amount of data it must process, or damage to code which is the result of other maintenance efforts on other parts of the system. Difficulties encountered during software maintenance can be reduced significantly by the adoption and enforcement of appropriate standards and procedures during the development and maintenance of the software.

Establish a Software Maintenance Policy

The establishment of a software maintenance policy for an organization is a vital step in controlling software maintenance. A software maintenance policy should describe in broad terms the responsibilities, authorities, functions, and operations of the software maintenance organization. It should be sufficiently comprehensive to address any type of change to the software system and its environment, including changes to the hardware, software and firmware. To be effective, the policy should be consistently applied and must be supported and promulgated by upper management to the extent that it establishes an organizational commitment to software maintenance. A software maintenance policy should direct attention toward the need for greater discipline in software design, development, and maintenance.

The software maintenance policy must specifically address the need and justification for changes, the responsibility for making the changes, the change controls and procedures, and use of modern programming practices, techniques and tools, It should describe management's role and duties in regard to software maintenance and define the process and procedures for controlling changes to the software. Implementation of the policy has the effect of enforcing adherence to rules regarding the operating software and documentation from initiation through

completion of the requested change. Once this is accomplished, it is possible to establish the milestones necessary to measure software maintenance progress. Reviews and audits are required to ensure the plans are carried out. The key to controlling changes to a system is the centralization of change approval and the formal requesting of changes.

Everything done to software affects its quality. Thus, measures should be established to aid in determining which category of changes are likely to degrade software quality. Care must also be taken to ensure that changes are not incompatible with the original system design and intent. The degree to which a change is heeded and its anticipated use should be a major consideration. Consideration should also be given the cost/benefit of the change: "Would a new system be less expensive and provide better capabilities?". The primary purpose of change control is to assure the continued smooth functioning of the application system and the orderly evolution of that system. Therefore, the policies establishing change control should be clear, concise, well publicized, and strictly enforced.

Review and evaluate all requests for changes

All user and staff requests for changes to an application system (whether enhancements, preventive maintenance, or errors) should be requested in writing and submitted to the software maintenance manager. Each change request should include not only the description of the requested change, but a full justification of why that change should be made. These change requests should be carefully reviewed and evaluated before any actual work is performed on the system. The evaluation should take into consideration, among other things, the staff resources available versus the estimated workload of the request; the estimated additional computing resources which will be required for the design, test, debug and operation of the modified system; and the time and cost of updating the documentation. Flexibility should be built into the process with some delegation of authority to initiate critical tasks when necessary. However, each request should be reviewed and evaluated by either the software maintenance manager or a change review board.

Plan for, and schedule maintenance

The result of the review of all change requests should be the assignment of a priority to each request and the updating of a schedule for meeting those requests, In many ADP organizations, there are simply more work requests than staff resources to meet those requests. Therefore, all work should be scheduled and every effort made to adhere to the schedule rather than constantly changing course in response to the most visible crisis.

Restrict code changes to the approved work

In many cases, especially when the code was poorly designed and/or written, there is a strong temptation to change other sections of the code as long as the program has been "opened up." The software maintenance manager must monitor the work of the software maintenance staff, and ensure that only the authorized work is performed. In order to monitor maintenance effectively, all activities must be documented. This includes everything from the change request form to the final revised source program listing.

Permitting software maintenance staff to make changes other than those authorized can cause schedules to slip and may prevent other, higher priority work from being completed on time. It is very difficult to limit the work which is done on a specific program, but it is imperative to the overall success of the software maintenance function to do so.

Enforce documentation and coding standards

Proper and complete communication of necessary information between all persons who have worked, are currently working, and who will work on the system is essential. The most important media for this communication are the documentation and the source code.

It is not enough to simply establish standards for coding and documentation. Those standards must be continually enforced via technical review and examination of all work performed by the software maintenance staff. In scheduling maintenance, sufficient time should be provided to fully update the documentation and to satisfy established standards and guidelines before a new assignment is begun.

4.2 Improving Software Maintenance

Maintainability is the ease with which software can be changed to satisfy user requirements or can be corrected when deficiencies are detected. The maintainability of a system must be taken into consideration throughout the lifecycle of that system. If the software is designed and developed initially with maintenance in mind, it can be more readily changed without impacting or degrading the effectiveness of the system. This can be accomplished if the software design is flexible, adaptable, and structured in a modular, hierarchical manner. Input from management, users, and developers during the design phase is also essential to improving maintainability. Such input enables the system designers to gain a better understanding of what is needed.

Many techniques and aids exist to assist the system developer, but there has been little emphasis on aids for the maintainer. However, since the processes which occur in the maintenance phase are similar to those of the development phase, there is considerable overlap in the applicability of the development aids in the maintenance environment.

The philosophies, procedures, and techniques presented here

should be utilized throughout the lifecycle of a system in order to provide maintainable software. Software systems which were not developed using these techniques can benefit from their application during major maintenance activities. As a system is maintained, the maintainability of the system can be improved by applying this guidance to the parts of the system which are modified during the maintenance process. While the effect will not be as pronounced as when programs are "developed with maintenance in mind," future maintenance efforts can be made easier by utilizing these techniques to "maintain systems with future maintenance in mind."

4.2.1 Source code Guidelines

Source code guidelines and standards aid maintainability by providing a structure and framework within which systems can be developed and maintained in a common, more easily understood, manner. Guidelines should reflect the needs and environment of a specific organization and should be based on the following basic principles:

Single high-order language

Wherever possible, a single high order language (HOL) should be used. High order languages resemble English and are generally, self-documenting. Since there are standards for most of the commonly used HOLs, it is easier to move a system written in an HOL from one environment to another.

Coding conventions

The first obstacle a maintainer must conquer is the code itself. A great deal of the source code written by developers and maintainers is not written with the future maintainer in mind. Thus, the readability of source code is often very poor. Source code should be self-documenting and written in a structured format.

Simple rules regarding the use of the language(s) and the physical formatting of the source code should be established. The following techniques can improve program readability and should be used as the basis for a code standard.

- Keep it simple. Complicated, fancy, exotic, tricky, confusing, or "cute" constructions should be avoided whenever a simpler method is available.

- Indentation, when properly utilized between sections of code, serves to block a listing into segments. Indentation and spacing are both ways to show subordination.

- Extensively comment the code with meaningful comments. Do not comment for comment's sake. Rather, comment in order to communicate to subsequent maintainers not only what was done and how it was done, but why it was done in this manner.

- Use meaningful variable names which convey both what the data item is and why it is used.

- Similar variable names should be avoided.

- Parameters should be used to pass data values between routines in a program.

- When numerals are used, they should be placed at the end of the variable name. Numbers used as program tags or labels should be sequential.

- Logically related functions should be grouped together in the same module or set of modules. To the extent possible, the logic flow should be from top to bottom of the program.

- Avoid non-standard feature of the version of the programming language being used.

Structured, modular software

A structured program is constructed with a basic set of control structures or modules which each have one exit and one entry point. Structured programming techniques are well-defined methods which incorporate top-down design and implementation and strict use of structured programming constructs. Whether the strict definition, or a more general approach (which is intended to organize the code and reduce its complexity) is used, structured programing has proven to be useful in improving the maintainability of a system.

A program comprised of small, hierarchical units or sets of routines, where each performs a particular, unique function, is said to be modular. Modularity is not merely program segmentation. Modules should be constructed using the following basic design principles:

- Modules should perform only one principal function.

- Interaction between modules should be minimal.

- Modules should have only one entry and one exit point.

Standard data definitions

It is imperative that a standard set of data definitions be developed for a system. These data definitions, which may be collected in a data dictionary, should define the name, physical attributes, purpose, and content of each data element utilized in the system. These names should be as descriptive and meaningful M possible. If this is consistently and correctly done, the task of reading and understanding each module and ensuring correct communication between each module is greatly simplified.

Well-commented code

The purpose of comments is to convey information needed to understand the process and the reasons for implementing it in that specific manner, not how it is being done. Good commentary increases the intelligibility of source code. In addition to making programs more readable, comments serve two other vital purposes. They provide information on the purpose and history of the program, its origin (the author, creation and change dates), the name and number of subroutines, and input/output requirements and formats. They also provide operation control information, instructions, and recommendations to help the maintainer understand aspects of the code that are not clear. Comments are often the primary form of documentation.

Compiler extensions

The use of non-standard features of a compiler can have serious effects on the maintainability of a system. If a compiler is changed, or the application system must be transported to a new machine, there is a very great risk that the extensions of the previous compiler will not be compatible with the new compiler. Thus, it is best to refrain from language extensions and to stay in conformance with the basic features of the language. If it is necessary to use a compiler extension, its use should be well-documented.

4.2.2 Documentation Guidelines

The documentation of a system is essential to good maintenance and should start with the original requirements and design specifications and continue throughout the lifecycle of the system. The documentation must be planned so a maintainer can quickly find the needed information. Documentation should support the useable transfer of pertinent information and should include instructions on what information must be provided, how it should be structured, and where the information should be kept.

A number of methodologies and guidelines exist which stress differing formats and styles. While preference may differ on which methodology to use, it is important to adopt a documentation standard and to then consistently enforce adherence to it for all software projects. In establishing documentation guidelines and standards, keep in mind that the Purpose is to communicate necessary, critical information, not to communicate all information.

The key to successful documentation is that not only must the necessary information be recorded, it must be easily and quickly retrievable by the maintainer. On-line documentation which has controlled access and update capabilities is the best form of documentation for the maintainer. If the documentation cannot be kept on-line, a mechanism must exist to permit access to the hard-copy documentation by the maintainer at any time.

Basically, the documentation, standards should require the

inclusion of all pertinent material in a documentation folder or notebook. There should be a requirement to complete and/or update documentation before new work assignments are begun. If documentation guidelines, or any other software guidelines or standards, are to be effective, they must be supported by a level of management high enough within the organization to ensure enforcement by all who use the software or are involved with software maintenance.

4.2.3 Coding And Review Techniques

The techniques listed in this section have been found to be very effective in the generation of maintainable systems. Not all techniques are generally applicable to all organizations, but it is recommended that they be considered.

Top down/bottom up approach

A top-down design approach (development or enhancements) involves starting at the macro or overview level and successfully breaking each program component or large, complex problem into smaller, less complicated segments. These segments are then decomposed into even smaller segments until the lowest level module of the original problem is defined for each branch in the logic flow tree.

The bottom-up design approach begins with the lowest level of elements. These are combined into larger components which are then combined into divisions, and finally the divisions are combined into a program. A bottom-up approach emphasizes designing the fundamental or "atomic" level modules first and then using these modules as building blocks for the entire system.

In most cases a combination of top-down and bottom-up approaches should be utilized to develop a clear, concise, maintainable system.

Peer reviews

Peer review is a quality assurance method in which two or more programmers review and critique each other's work for accuracy and consistency with other parts of the system. This type of review is normally done by giving a section of code developed by one programmer to one or more other peer programmers who are charged with identifying what they consider to be errors and potential problems. It is important to establish and to keep clearly in the participants' minds that the process is not an evaluation of a programmer's capabilities or performance. Rather, it is an analysis and evaluation of the code. As stated in the name, such reviews are performed on a peer basis (programmer to programmer) and should never be used as a basis for employee evaluation. Indeed, project managers should not, if possible, be involved in the peer reviews.

Inspections

Inspection refers to a formal evaluation technique employed to identify discrepancies and to measure quality (error content) of an application system throughout the software lifecycle. The inspection generally begins with an overview that identifies procedural logic, paths, and interdependencies of the plan or deliverables. Next, the areas to be scrutinized are flagged and examined in detail to detect faults, violations of development and maintenance standards, and other problems. A report is then produced which identifies errors uncovered and requirements for error correction.

The responsibilities and activities of the inspection team, and a checklist which identifies the deliverables and materials to be examined are prepared in advance. This checklist is updated as uncovered errors are resolved and new errors found. Inspections are generally performed by a team composed of a team leader and other persons who have an indepth knowledge of the system design and functions, an understanding of the specific program areas to be inspected, and an ability to determine and establish test criteria. The team may, but normally does not, include the author of the plans, programs, or deliverables. A follow-up inspection is performed to ensure that each error has been corrected successfully. While inspections are more formal and generally require greater effort than walkthroughs, their benefits lie in the capacity to produce repeatable results at specified checkpoints.

Walkthroughs

Walkthroughs of a proposed solution or implementation of a maintenance task can range from informal to formal, unstructured to structured, and simple to full-scale. In its simplest form, a walkthrough can be two maintainers sitting down and discussing a task which one of them is working on. In its more complex forms, there may be a structured agenda, report forms, and a recording secretary. The goal is an open, frank dialogue which results in the refinement of good ideas and the changing or elimination of bad ones. Managers may or may not participate in walkthroughs.

4.2.4 Controlling Change

Change control is necessary to ensure that all software maintenance requests are handled accurately, completely, and in a timely manner. It helps assure adherence to the established standards and performance criteria for the system and facilitates communication between the software maintenance team members and the software maintenance manager.

There should be a centralized approval point for all software maintenance projects. This may be the software maintenance project manager or, for larger systems or organizations, a review board. The centralized approval process will enable one person or group of persons to have knowledge of all the requested and actual work being performed on the system. If this is not done, there is the likelihood that two or more independent changes to the system will be in conflict with one another and as a result, the system will not function properly.

Additionally, different users will often request the same enhancements to a system but will have small differences in the details. By coordinating these requests, details can be combined and the total amount of resources required can be reduced.

Change request

There must be a formal, well-defined mechanism for initiating a request for changes or enhancements to a system. All changes considered for a system should be formally requested in writing. These requests may be initiated by the user or maintainer in response to discovered errors, new requirements, or changing needs. Procedures may vary regarding the format of a change request, but it is imperative that each request be fully documented in writing so it can be formally reviewed. Change requests should be carefully evaluated by the project manager or a change review board and decisions to proceed should be based on all the pertinent areas of consideration (probable effects on the system, actual need, resource requirements vs resource availability, budgetary considerations, priority, etc.). The decision and reasons for the decision should be recorded and included in the permanent documentation of the system.

Review and approval

Review and approval is the process of confirming that a software system meets the requirements and design specifications in the operational environment. It is a process which assures that the Integral parts of the system perform according to the specifications.

Prior to installation, each change (correction, update, or enhancement) to a system should be formally reviewed. In practice this process ranges from the review and sign-off by the project manager or user, to the convening of a change review board to formally approve or reject the changes. The purpose of this process is to ensure that all of the requirements of the change request have been met; that the system performs according to specifications, that the changes will not adversely impact the rest of the system or other users; that all procedures have been followed and rules and guidelines adhered to; and that the change is indeed ready for installation in the production system. All review actions and findings should be added to the system documentation folder.

Code Audit

Code review or audit is a procedure used to determine how well code adheres to established coding standards and practices and to the design specifications. The primary objective of code audits is to guarantee a high degree of uniformity across the software. This becomes a critical factor when someone other than the original developer must understand and maintain the software. Audits are also concerned with such program elements as commentary, labeling, paragraphing, initialization of common areas, and naming conventions. The audit should be performed by someone other than the original author.

4.2.3 Testing Standards and Procedures

Testing standards and procedures should define the degree and depth of testing to be performed and the disposition of test materials upon successful completion of the testing. Whenever possible, the test procedures and test data should be developed by someone other than the person who performed the actual maintenance on the system.

Testing is a critical component of software maintenance. As such the test procedures must be consistent and based on sound principles. The test plan should define the expected output and test for valid, invalid, expected, and unexpected cases. The test should examine whether or not the program is doing what it is supposed to do. The goal of testing is to find errors, not to prove that errors do not exist.

4.3 Managing Software Maintainers

Management is clearly one of the most Important factors in improving the software maintenance process. Management must examine how the software is maintained, exercise control over the process, and ensure that effective software maintenance techniques and tools are employed. In order to maintain control over the software maintenance process and to ensure that the maintainability of a system does not deteriorate, it is important that software maintenance be anticipated and planned for.

The effective use of good management techniques and methodologies in dealing with scheduling maintenance, negotiating with users, coordinating the maintenance staff, and instituting the use of the proper tools and disciplines is essential to a successful software maintenance effort. Software maintenance managers are responsible for making decisions regarding the performance of software maintenance; assigning priorities to the requested work; estimating the level of effort for a task; tracking the progress of work; and assuring adherence to system standards in all phases of the maintenance effort. A software maintenance function has the same organizational needs and managerial problems as any other function.

Selecting the proper staff for a software maintenance project is as important as the techniques and approaches employed. While separate staffs for maintenance and development can improve the effectiveness of both, the realities of size, organization, budget, and staff ceilings often preclude the establishment of separate maintenance and development staffs.

Management must apply the same criteria to the maintainers that are applied to software and systems designers or other highly sought after professional positions. If an individual is productive, consistently performs well, has a good attitude, and displays initiative, it should not matter whether the project is development or maintenance. Three major psychological factors can impact the attitude, morale, and general performance of an

individual:

- the work must be considered worthwhile by a set of values accepted by the individual, as well as by the standards employed by the organization.

- the individual must feel a responsibility for his or her performance. There is a need to feel personally accountable for the outcome of an effort.

- the Individual must be able to determine on a regular basis whether or not the outcome of his or her effort is satisfactory.

It is essential that work assignments offer growth potential. Continuing education is required at all levels to ensure that not only the maintainers, but the users, managers, and operators have a thorough understanding of software maintenance. Training should include: programming languages, standards and guidelines, operating systems, and utilities. Figure 3 outlines some points to keep in mind when staffing and managing a software maintenance function.

5. SYSTEM MAINTENANCE VS SYSTEM REDESIGN

Although maintenance is an ongoing process, there comes a time when serious consideration should be given to redesigning a software system. A major concern of managers and software engineers is how to determine whether a system is hopelessly flawed or whether it can be successfully maintained. The costs and benefits of the continued maintenance of software which has become error-prone, ineffective, and costly must be weighed against that of redesigning the system.

When a decision has been reached to redesign or to stop supporting a system, the decision can be implemented in a number of ways. Support can simply be removed and the system can die through neglect; the minimum support needed to keep it functioning may be provided while a new system is built; or the system may be rejuvenated section by section and given an extended life. How the redesign is affected depends on the individual circumstances of the system, its operating environment, and the needs of the organization it supports.

While there are no absolute rules on when to rebuild rather than maintain the existing system, some of the factors to consider in weighing a decision to redesign or maintain are listed in figure 4. These characteristics are meant to be general "rules of thumb" which can assist a manager in understanding the problems in maintaining an existing system and in deciding whether or not it has outlived its usefulness to the organization. The greater the number of characteristics present, the greater the potential for redesign.

Frequent System Failures

A system which is in virtually constant need of corrective maintenance is a prime candidate for redesign. As systems age and additional maintenance is performed, many become increasingly fragile and susceptible to changes. The older the code the more likely frequent modifications, new requirements, and enhancements will cause the system to break down.

An analysis of errors should be made to determine whether the entire system is responsible for the failures, or if a few modifications or sections of code are at fault. If the latter is found to be the case, then redesigning those parts of the system may suffice.

Code Over 7 Years Old

The estimated lifecycle of a major application system is 7-to-10 years. Software tends to deteriorate with age as a result of numerous fixes and patches. If a system is more than 7 years old, there is a high probability that it is outdated and expensive to run. A great deal of the code in use today falls into this category. After 7-to-10 years of maintenance, many systems have evolved to where additional enhancements or fixes are very timeconsuming to make. A substantial portion of this code is probably neither structured, nor well-written, While this code was adequate and correct for the original environment, changes in technology and applications may have rendered it inefficient, difficult to revise, and in some cases obsolete. On the other hand, if the system was designed and developed in a systematic, maintainable manner, and if software maintenance was carefully performed and documented using established standards and guidelines, it may be possible to run it efficiently and effectively for many more year's.

Overly Complex Program Structure and Logic Flow

"Keep it simple" must be the "golden rule" of all programming standards and guidelines. Too often, programmers engage in efforts to write a section of code in the least number of statements or utilizing the smallest amount of memory possible. This approach to coding usually results in complex code which is virtually incomprehensible. Poor program structure contributes to complexity. If the system being maintained contains a great deal of this type of code and the documentation is also severely deficient, it is a candidate for redesign.

Complexity also refers to the level of decision making present in the code. The greater the number of decision paths, the more complex the software is likely to be. Additionally, the greater the number of linearly independent control paths in a program, the greater the program complexity. Programs characterized by some or all of the following attributes are usually very difficult to maintain and are candidates for redesign:

- excessive use of DO loops

- excessive use of IF statements

- unnecessary GOTO statements

- embedded constants and literals

- unnecessary use of global variables

- self-modifying code

- multiple entry or exit modules

- excessive interaction between modules

- modules which perform same or similar functions.

Code Written for Previous Generation Hardware

Few industries have experienced as rapid a growth as the computer industry, particularly in the area of hardware. Not only have there been significant technological advances, but, the cost of hardware has decreased dramatically during the last decade. This phenomenon has generated a variety of powerful hardware systems. Software written for earlier generations of hardware is often inefficient on newer systems. Attempts to superficially modify the code take advantage of the newer hardware is generally ineffective, time-consuming and expensive.

Running in Emulation Mode

One of the techniques used to keep a system running on newer hardware is to emulate the original hardware and operating system. Emulation refers to the capability of one system to exhibit behavior characteristic of another machine. In effect, it makes the host machine imitate the emulated machine. Emulation is normally used when resources are not available to convert a system, or the cost of doing so would be prohibitive. It frequently prevents utilization of the total capabilities and full power of the newer system. Emulated systems run a very fine line between functional usefulness and total obsolescence.

Very Large Modules or Unit Subroutines

"Mega-systems" which were written as one or several very large programs or sub-programs (thousands or tens-of-thousand of lines of code per program) can be extremely difficult to maintain. If the large modules can be restructured and divided into smaller, functionally related sections, the maintainability of the system will be improved.

Excessive Resource Requirements

An application system which requires a great deal of CPU time, memory, storage, or other system resources can place a very serious burden on all ADP users. Issues which must be

addressed, include whether it is cheaper to add more computer power or to redesign and reimplement the system, and whether redesign will reduce the resource requirements.

Hard-Coded Parameters Which Are Subject To Change

Many older systems were designed with the values of parameters used in performing specific calculations "hard coded" into the source code rather than stored in a table or read in from a data file. When changes in these values are necessary, (withholding rates, for example) each program in the system must be examined, modified and recompiled as necessary. This is a time-consuming, error prone process which is costly both in terms of the resources necessary to make the changes and the delay in getting the changes installed.

Whenever possible, the programs should be modified to handle the input of parameters in a single module or to read the parameters from a central table of values.

Difficulty in Keeping Maintainers

Programs written in low level languages, particularly assembler, require an excessive amount of time and effort to maintain. Generally, such languages are not widely taught or known. Therefore, it will be increasingly difficult to find maintainers who already know the language.

Seriously Deficient Documentation

Too often, documentation ranges from nonexistent to out-of-date. Even if the documentation is good when delivered, it often steadily and rapidly deteriorates as the software is modified. In some cases, the documentation is up-to-date, but still not useful. This can result when the documentation is produced by someone who does not understand the software or what is needed.

The worst documentation is that which is well-structured and formatted but which is incorrect or outdated. If there is no documentation, the maintainer will be forced to analyze the code in order to try to understand the system. If the documentation is physically deteriorated, the maintainer will be skeptical of it and verify its accuracy. If it looks good on the surface, but is technically incorrect, the maintainer may mistakenly believe it to be correct and accept what it contains. This will result in serious problems over and above those which originally necessitated the initial maintenance.

Missing or Incomplete Design Specifications

Knowing "how and why" a system works is essential to good maintenance. If the requirements and design specifications are missing or incomplete, the task of the maintainer will be more difficult. It is very important for the maintainer to not only understand what a system is doing, but how it is implemented, and why it was designed.

GLOSSARY

ADAPTIVE MAINTENANCE: Any effort which is initiated as a result of changes in the environment in which a software system must operate.

APPLICATION UTILITY LIBRARIES: See SOFTWARE LIBRARY.

APPLICATION SOFTWARE: Software specifically produced for the functional use of a computer system, for example payroll, general ledger, inventory control, human resources management.

BASELINE: A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development or maintenance, and that may be changed only through formal change control procedures.

BOTTOM-UP APPROACH: An approach that starts with the lowest level software components of a hierarchy and proceeds through progressively higher levels to the top level component.

CHANGE CONTROL: The process by which a change is proposed, evaluated, approved or rejected, scheduled, and tracked.

CHANGE REVIEW BOARD: The authority responsible for evaluating or disapproving proposed engineering changes, and ensuring implementation of the approved changes. Also referred to as the Configuration Control Board.

CHIEF PROGRAMMER: The leader of a chief programmer team. A senior-level programmer whose responsibilities include producing key portions of the software assigned to the team, coordinating activities of the team. reviewing work of the other team members, and having overall technical understanding of the software being developed or maintained.

CHIEF PROGRAMMER TEAM: A software development or maintenance group that employs support procedures designed to enhance group communication and to make optimum use of each member's skills,

CODE AUDIT: An independent review of source code by a person, team, or tool to verify compliance with software design, programming, and documentation standards.

CODE INSPECTION: The use of a formal set of procedures which are used to examine and measure the quality (error content) of the software.

COHESION: Cohesion refers to the degree to which the functions or processing elements within a module are related or bound together.

COMPILER: A computer program used to translate a high order language program into executable machine instructions.

COMPILER EXTENSION: Features of a programming language which are

not included in the standard features (e.g., ANSI standard) of that language but are accepted and compiled by a specific compiler.

COMPLEXITY: The degree of complication of a system or system component, determined by such factors as the number and intricacy of interfaces, the number and intricacy of conditional branches, the degree of nesting, the types of data structures, and other system characteristics.

CORRECTIVE MAINTENANCE: Changes to a software system which are necessitated by actual errors (induced or residual) in a system.

COUPLING: The degree that modules are dependent upon each other in a computer program.

DESIGN REVIEW: The formal review of an existing or proposed design for the purpose of detection and remedy of design deficiencies and/or errors, and for the identification of possible improvements.

EMULATION: The imitation of all or part of one computer system by another so the imitating system accepts the same data, executes the same programs, and achieves the same results as the Imitated system.

FIRM WARE: Computer programs and data loaded in a class of memory that cannot be dynamically modified by the computer during processing.

HIGH ORDER LANGUAGE (HOL): A programming language that does not reflect the logical structure of any one given computer or class of computers, for example, COBOL, FORTRAN, and PL/I.

LIBRARIAN: See SOFTWARE LIBRARIAN.

LIBRARY: See SOFTWARE LIBRARY.

LIFECYCLE: See SOFTWARE LIFECYCLE.

MAINTAINABILITY: The ease with which software can be maintained, for example, enhanced, adapted, or corrected to satisfy specified requirements.

MAINTENANCE: See SOFTWARE MAINTENANCE.

MODULAR: A program compiled of small, hierarchical units or sets of routines, where each performs a particular unique function, is said to be modular.

MODULE: A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading, for instance, the input to, or output from, an assembler, compiler, linkage editor, or executive routine.

PEER REVIEW: A quality assurance method in which two or more peer programmers review and critique each other's work for accuracy

and consistency with other parts of the system.

PERFECTIVE MAINTENANCE: All changes, insertions, deletions, modifications, extensions, and enhancements which are made to a system to meet the evolving and/or expanding needs of the user.

PROGRAM: A sequence of instructions suitable for processing by a computer.

REGRESSION TESTING: Rerunning test cases which a program has previously executed correctly to detect errors created during software maintenance.

SOFTWARE: Computer programs, procedures, rules, and possibly associated documentation and data pertaining to the operation of a computer system.

SOFTWARE LIBRARIAN: The person responsible for establishing, controlling, and maintaining a software library.

SOFTWARE LIBRARY: A controlled collection of software and related documentation designed to aid in software development, use, or maintenance.

SOFTWARE LIFECYCLE: The period of time beginning when a software product is conceived and ending when the product is no longer available for use. The software lifecycle is typically broken into phases, such as requirements, design, implementation, testing, and operations and maintenance.

SOFTWARE MAINTENANCE: The performance of those activities required to keep a software system operational and responsive after It is accepted and placed into production.

SOFTWARE TOOL: A computer program used to help develop, test, analyze, or maintain another computer program or its documentation.

SOURCE CODE: The program instructions written in a programming language,

STRUCTURED DESIGN: A disciplined approach to software design that adheres to a specified set of rules based on principals.

STRUCTURED PROGRAM: A program constructed of a basic set of control structures, each one having one entry point and one exit point. Less formally, any program which conforms to some disciplined approach intended to control the design, format, and logic structure of the program.

TESTING: Examining the behavior of a program by executing the program on sample data sets.

TOOL: See SOFTWARE TOOL.

TOP-DOWN APPROACH: An approach that starts with the highest level component of a hierarchy and proceeds through progressively lower

levels.

VALIDATION: Determination of the correctness of the final program or software produced from a development project with respect to the user needs and requirements.

VERIFICATION: The demonstration of consistency, completeness, and correctness of the software at each stage and between each stage of the development lifecycle.

VV&T: Validation, verification, and testing; used as an entity to define a procedure of review, analysis, and testing throughout the software lifecycle to discover errors, determine functionality, and ensure the production of quality software.

WALKTHROUGH: A manual analysis technique in which the module author describes the module's structure and logic to an audience of colleagues.

NOTE: Most of the definitions in this glossary appear in one or more of the following:

1. NBS Special Publication 500-106, "Guidance on Software Maintenance," by R. Martitt and W. Osborne, December, 1983.

2. IEEE Computer Society, IEEE Std 729-1983, "IEEE Standard Glossary of Software Engineering Terminology," February, 1983.

3. Federal Information Processing Standard (FIPS) 11-2, :Guideline: American National Dictionary for Information Processing Systems," American National Standards Committee X3, Information Processing Systems, X3/TR-1-82, 1982.

APPENDIX I

Supporting ICST Documents

[FIPS38] "Guidelines for Documentation of Computer Programs and Automated Data Systems," FIPS PUB 38, 1976.

[FIPS64] "Guidelines for Documentation of Computer Programs and Automated Data Systems for the Initiation Phase," FIPS PUB 64, 1979.

[FIPS101] "Guideline for Lifecycle Validation, Verification, and Testing of Computer Software," FIPS PUB 101, 1983.

[NBS56] NBS Special Publication 500-56 "Validation, Verification, and Testing for the Individual Programmer," M. Branstad, J. Cherniavsky, and W. Adrion, 1980.

[NBS75] NBS Special Publication 500-75 "Validation, Verification, and Testing of Computer Software," W. Adrion, M. Branstad, and J. Cherniavsky, 1981.

[NBS87] NBS Special Publication 500-87 "Management Guide to Software Documentation," A. Neumann, 1982.

[NBS88] NBS Special Publication 500-88 "Software Development Tools." R. Houghton, Jr., 1982.

[NBS93] NBS Special Publication 500-93 "Software Validation, Verification, and Testing Technique and Tool Reference Guide." P. Powell, Editor, 1982.

[NBS106] NBS Special Publication 500-106 "Guidance on Software Maintenance," R, Martin and W. Osborne, 1983.

Notes:$I 1. Subsequent NBS documents will include guidance on acceptance testing and other software engineering topics.

2. FIPS Guidelines documents may be ordered from:

National Technical Information Service

5285 Port Royal Road

Springfield, VA 22161

(703) 487-4650

3. NBS Special Publications may be ordered from:

Superintendent of Documents

U.S. Government Printing Office

Washington, DC 20462

(202) 783-3238

APPENDIX II

Suggested Additional Reading Material

[COUG82] D. J. Couger and M. A. Colter, "Effect of Task Assignments on Motivation of Programmers and Analysts," research report, University of Colorado, 1982.

[DONA80] J. Donahoo and D. Swearinger, "A Review of Software Maintenance Technology," Rome Air Development Center, RADC-TR-80-13, February 1980.

[GLAS79] R. L Glass, Software Reliability Guidebook Prentice-Hall, Englewood Cliffs, NJ, 1979.

[GLAS81] R. L. Glass and R. A. Noiseux, Software Maintenance Guidebook, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[LIEN80] B. P. Lientz and E. B. Swanson, Software Maintenance Management, Addison-Wesley, Reading, MA, 1980.

[MART83] J. Martin, C. McClure, Software Maintenance--The Problem and Its Solutions, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[MCCL81] C. L. McClure, Managing Software Development and Maintenance, Van Nostrand Reinhold, NY, 1981.

[PARI80] G. Parikh, editor, Techniques of Program and System Maintenance, Ethnotech, Lincoln, NE, 1980.

[PARI83] G. Parikh, N. Zvegintzov, Tutorial on Software Maintenance, IEEE Computer Society Press, Silver Spring, MD, 1983.

[PERR81] W. E. Perry, Managing System Maintenance, Q.E.D. Information Sciences, Inc., Wellesley, MA. 1981.

[PRES82] R. Pressman, Software Engineering: A Practitioner's Approach, McGraw Hill, New York, 1982.

NBS Technical Publications

Periodicals

Journal of Research--The Journal of Research of the National Bureau of Standards reports NBS research and development in those disciplines of the physical and engineering sciences in which the Bureau is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Bureau's technical and scientific programs. As a special service to subscribers the issue contains complete citations to all recent Bureau publications in both NBS and non-NBS media. Issued six times a year.

Nonperiodicals

Monographs--Major contributors to the technical literature on various subjects related to the Bureau's scientific and technical activities.

Handbooks--Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications--Include proceedings of conferences sponsored by NBS, NBS annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

Applied Mathematics Series--Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others

engaged in scientific and technical work.

National Standard Reference Data Series--Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a world-wide program coordinated by NBS under the authority of the National Standard Data Act (Public Law 90-396). NOTE: The Journal of Physical and Chemical Reference Data (JPCRD) is published quarterly for NBS by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements are available from ACS, 1155 Sixteenth St.. NW, Washington, DC 20056.

Building Science Series--Disseminates technical information developed at the Bureau on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes--Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NBS under the sponsorship of other government agencies.

Voluntary Product Standards--Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NBS administers this program as a supplement to the activities of the private sector standardizing organizations.

Consumer Information Series--Practical information, based on NBS research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping todays technological marketplace. Order the above NBS publication from: Superintendent of Documents Government Printing Office, Washington, DC 20402. Order the following NBS publications--FIPS and NBSIR's--from the National Technical Information Service. Springfield, VA 22161.

Federal Information Processing Standards Publications (FIPS PUB)--Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NBS pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of title 15 CFR (Code of Federal Regulations).

NBS Interagency Reports (NBSIR)--A special series of interim or

final reports on work performed by NBS for outside sponsors (both government and non government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Service, Springfield, VA 22161, in paper copy or microfiche form.

11/18/02

Help Desk Call Averages
Contract No. DTRS57-97-C-00107

Below are the approximate average Helpdesk Calls received per month over the past 12-month period broken down by functional area:

Function	Average Monthly Calls
Software	223
Hardware	164
EMAIL	91
Reset Password	49
Network	40
Others	74
Total	640

Table of Contents

IADPSS Monthly Progress Report
XXXXX, 2002 to XXXXX, 2002

Administrative Applications

Manager: XXXXX, Volpe Initiator: XXXXX
XXXXX, 2002 to XXXXX, 2002

Activities for Reporting Month

In the **Production and Maintenance Area**, general production support work continued with the following notable special maintenance activities.

Support was provided for processing of **XXXXXXXX** and **XXXXXX** data. This included resolving database and systems issues, and programming several ad-hoc reports. Normal batch processing of XXXXX and XXXX feeds into XXXX occurred.

The XXXXX database was purged of XXXXX codes.

The **XXXXX**, and **XXXXX Systems** experienced no problems during the period.

The **XXXXX System** users requested and received eight special reports.

The **XXXXX System** users had trouble accessing the data maintenance system. The problem was resolved by rebooting their PC.

The following notable **New Work Task** activities took place during the reporting month:

In the **XXXXX** area, weekly Work Group and bi-weekly Checkpoint meetings continued. Work on outstanding tasks associated with the contracts implementation continued. For more information on XXXXX specific issues, see the *Strategic Systems Planning* section of this report.

In the **XXXXX System** area, working sessions continued to complete the tasks associated with developing the XXXXX feed to XXXXX.

Analysis XXXX regarding the conversion of the XXXX database to the XXXXX platform and database structures.

For more information, see the *Strategic Systems Planning* section of this document.

In the **XXXXX** area, support was provided to the ongoing validation effort. Daily comparisons of XXXXX-Incremental data continued and were compiled, analyzed and presented to XXXXX.

The effort to add new fields to the data extracts for the **XXXX** area continued. These additional data fields are required to support the informational data requirements of XXXXX. An initiative to re-organize and finalize the data extracts for the XXXXX area was implemented. The enhanced format of the extracts will allow for more efficient management when the data staging area is released for XXXXX.

For more information, see the *Strategic Systems Planning* section of this document.

Activities for Next Month

During the next month, the following activities are planned in addition to performing routine production and maintenance support.

Support will be provided for month-end XXXXX and opening of XXXXX and the **XXXX** for the following month.

Normal maintenance activities are planned for **XXXXX,** and **XXXX Systems** for next period.

Normal maintenance activities are planned for the **XXXXX System**.

In the **XXXXX Systems** area, the subcommittee working sessions will continue to address the issues related to completing the XXXX task.

Programming of the XXXXX feed to XXXX will be tested.

Analysis of all legacy systems will continue to identify interfaces with the XXXX database.

Analysis will continue regarding the conversion of the XXXX database for the XXXXX project. The XXXX data migration will require XXXXXX.

In the **XXXX** area, the effort to monitor and analyze the XXXXX will continue. XXXXXX.

For more information about the XXXXX tasks, see the *Strategic Systems Planning* section of this document.

Special Interest Items/Issues

None.

AIS Security Program

Manager: XXXXX, Volpe Initiator: XXXXX
XXXXX, 2002 to XXXXX, 2002

Activities for Reporting Month

Data documentation for the XXXX Database including data XXXXX was created. In line with the data documentation, a newly created XXXXX was documented and proposed.

XXXXX.

The Center-wide XXXXXX

A complete listing of all outstanding operating system and software patches was supplied to XXXXX. XXXXX.

XXXXX efforts were once again conducted to provide security log reviews XXXXX.

A design was developed and an implementation process was performed to XXXXXX.

Remediation and configuration changes were applied to several servers as part of the continuing efforts to XXXXXX.

Several planning sessions were held regarding the XXXXXX

All XXXXX, XXXXX Web sites were reviewed for compliance to DOT web site regulations.

Discussions were initiated for security compliance, SSP preparation, and Certification efforts for XXXXX manager.

Planning meetings were held with the Applications Development team for the XXXX Project. Security Planning will be a XXX on-going component of this effort.

Several Vendor sessions were conducted XXXXX. These included presentations by Veritas, Symantec, RSA, Vericept and GTSI.

Approaches to meeting the Federal and DOT requirement for Awareness and Training were discussed. XXXXXX.

Research was conducted and results provided to XXXX regarding privacy statements on individual web pages.

XXXXXX.

Discussions were held regarding XXXXXXX.

Security issues related to password XXXXXX.

XXXXXXX.

The XXXXXX.

Windows 2000 Migration meetings were attended. XXXXXXXX.

A number of project/program meetings were attended for the purpose of providing input regarding security issues. XXXXXXXs.

Monitoring of log files continued on the firewall XXXXXXX.

XXXXXX.

XXXXXXX.

Discussions with Symantec provided updated information about their current firewall and security product offerings.

Activities for Next Month

Work will continue on updating all Volpe Center XXXXXXX.

ISS scans for vulnerabilities of both new systems being deployed and existing systems connected to the Volpe LAN will continue.

Reviews of current technology will be conducted to ensure a state-of-the-art capability is maintained for addressing security issues within the Volpe Center environment.

XXXXXXX

Efforts will be conducted to address any XXXXXXXX.

Reviews of logs will be conducted.

Ad hoc tasks will be addressed.

Special Interest Items/Issues

XXXXXXXX.

Facility Operations

Manager: XXXXXXXX, Volpe Initiator(s): XXXXXX
XXXXXX, 2002 to XXXXXX, 2002

Activities for Reporting Month

The following activities were completed or continue to be worked on by Facility Operations personnel during XXXXX.

The Data Entry team in **User Services** provided on-going support to XXXXX during this recording period.

The XXXXX group in User Services provided on-going support to the XXXXX during this recording period.

User Services provided Systems Administration support to the XXXXX during this recording period.

During XXXXX, the on-going support of storage management activities continued through general OpenVMS cluster disk maintenance and media management in the **XXXXX** area. The series of tasks in support of the Data Center OpenVMS cluster storage facility also included monitoring disk capacities to avoid device saturation, performing any/all monthly housekeeping on VCS media and performing general disk housekeeping on the XXXXX and XXXXX devices.

Through the month, the **OpenVMS Systems Managers** performed various systems administration tasks supporting XXXXXX.

The OpenVMS Systems Managers, in concert with the Systems Administration Team, performed UNIX system administration tasks that included XXXXX.

The OpenVMS Systems Managers completed work on the implementation of the XXXXX.

The OpenVMS Systems Managers, in concert with the Systems Administration Team, continued to work with XXXXX evaluation installation.

The OpenVMS Systems Managers, in concert with the Systems Administration Team, installed XXXXX.

The OpenVMS Systems Managers assisted Administrative Applications with modifications to the XXXXX.

The OpenVMS Systems Managers replaced and rebuilt the system disk on XXXXX.

The OpenVMS Systems Managers continued work on the XXXXXX provided the appropriate document to assist in determining what level of service could potentially be provided.

Oracle Initiatives

- XXXXX
- XXXXX
- XXXXXXXX

- During the month of XXXXX **the System Administration Team (UNIX and Windows NT/2000)** continued it's ongoing support and maintenance of all servers and associated services. XXXXXXXXX

Project Work:

Facility Operations

Manager: XXXXXXXX, Volpe Initiator(s): XXXXXX
XXXXXX, 2002 to XXXXXX, 2002

UNIX project work in the month of XXXXX included resolving a printer queue issue on the XXXXX.

The team worked with XXXXX.

Windows NT/2000 efforts included troubleshooting specific networking and connectivity problems with the *XXXXXX*.

Extensive support was provided to the XXXXXXX.

- **Network Engineering** completed the following XXXXX

- Network Engineering completed the following XXXXXX

During the UPS test late in the month of XXXXX, a failed UPS battery was identified in the Data Center; this UPS supports the XXXXXX

Network Engineering worked with XXXXX.

A XXXX Client has been XXXX.

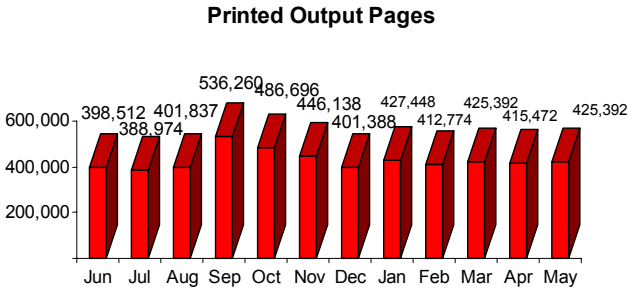
Activities for Next Month

- **Network Engineering** is preparing work on the following VPN Implementations for XXXXXX

XXXXX.

The **System Administration Team (UNIX and Windows NT/2000)** and the **OpenVMS Systems Managers** will continue their on-going support and maintenance of all Data Center servers as well as their associated services.

Other:



XXXXXXXXX.

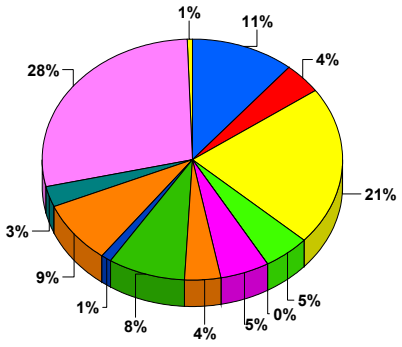
PC/Network Support

Manager: XXXXX, Volpe Initiator: XXXXXX
XXXXX, 2002 to XXXX, 2002

Activities for Reporting Month

The Exchange Mailbox totals for the current month are listed below:

	Additions	Deletions	Total
Mailboxes	28	3	1368
Remote Mail	3	0	289



Software	**260**
Hardware	**208**
EMAIL	**105**
Reset Password	**60**
Network	**37**
Others	**30**
Total Calls:	**700**

Exchange: Most of XXXXX was spent researching and planning the windows 2000 migration, specifically as it relates to the Exchange servers. XXXXXXX.

XXXXXXX.

Gateway Antivirus Activity Report

01-XXXXXXX-2002 00:00:00 - 01-XXXXXXX-2002 00:00:00

Message Summary

Messages Accepted	226306
Data Accepted (KB)	9181547
Messages Blocked By Subject	0
Messages Rejected	20
Messages Bounced	52
Messages Delivered	222745
Message Delivery Failures	11
Messages Completed	225793
Attachments Deleted	663

Infection Summary

Infections Ignored	0
Infections Repaired	5
Infections Deleted	124
Total	129
Infections Quarantined	0

Desktop: In XXXXX, the desktop technicians XXXXX.

In XXXXX, the desktop group continued to assist in the installation of IE XXXXX.

Special Interest Items/Issues

None.

Strategic Systems Planning

Manager: XXXXXVolpe Initiator: XXXXX
XXXXX, 2002 to XXXXX, 2002

Activities for Reporting Month

XXXXXX): The system was supported during the period without incident.

The XXXXX application re-hosting from the Open VMS operating system to the Tru64 Unix operating system continued. XXXXX.

The XXXX Area was used to populate an XXXX test database. XXXXX.

XXXXX: Several individual users with specific problems were assisted at their desktops. Several new accounts were created. XXXXX.

XXXXX: The development of a list of data errors that exist in XXXX is in-progress. This list will provide the basis for developing an XXXXX that will run daily and generate a diagnostic error report for use in correcting them.

Work also continued on generating required weekly XXXXX reports.

- XXXXX XXXXXXX

XXXXX XXXXX

XXXXX:

XXXXX Module: Meetings were held with XXXX to review the methods developed for production processing of the interface feeds to XXXXX. Several enhancements were made to the GUI XXXXX.

XXXXX Module: The XXXX instance was populated with sample data files. The actual steps for processing them have changed since the last time this was tested. All relevant processing scripts were accumulated and combined, and a production process solution was created for review by the XXXX group.

XXXXX: Members of the XXXXX team met and identified XXXX reports to use for conversion testing. XXXXX.

XXXXX: The XXXXXX team contacted the XXXXX staff responsible for XXXXX data extracts and is awaiting their response. No new activity occurred this month.

XXXXX: The XXXX software was received, and those portions XXXX were installed. XXXXX. XXXXXX software were installed successfully and the server was scheduled for the required ISS security scan.

Data validation activities continued, incorporating XXXXX along with the analysis of XXXXX Daily e-mails monitoring the differences between XXXXX-Incremental were analyzed and presented to XXXXXXX.

The planning effort for the design and implementation of a XXXX was completed.

XXXXXX: The XXXX Requirements and the XXXXXX Requirements were finalized and await submission to the XXXX team once the new XXXX plan is established.

Strategic Systems Planning

Manager: XXXXXVolpe Initiator: XXXXX
XXXXX, 2002 to XXXXX, 2002

The XXXXX processes were determined, refined, and draft documentation was generated. This included XXXXXXX.

XXXXXX: Technical and planning meetings were temporarily suspended.

XXXX Web: Both the XXXXX sites were monitored to ensure availability and reliability. Feedback for both was gathered and responses coordinated.

The evaluation of XXXXX for use with the Intranet and Internet continued.

Planning continued for an upgrade of the current XXXXXX servers.

XXXXX team members continued to participate in meetings related XXXXXX.

XXXXX: The XXXXXX was updated with new XXXXXXXXX.

XXXXXXs were launched on the Intranet.

Information on Technical Topics is in XXXXXXX.

A "XXXXXXXX.

XXXXXXX.

XXXXXXXXX.

Volpe Internet: Updates were made to several sections, including the XXXXXX. Work continued on XXXXX to meet Section 508 standards.

XXXXXXX.

Other Internet: The XXXXXX.

XXXXX team members supported a short-term effort related to the XXXXXXX

Activities for Next Month

XXXXXXX): Work will continue to establish additional fiscal year data in the test database XXXXXXX.

XXXXXX: Assistance to Desktop Support and User Administration will continue. Research will continue regarding the XXXXXX.

XXXXXX): Selected XXXXX reports will be identified for development.

Work will continue to resolve the XXXXXX.

XXXXXreports were selected for conversion to Crystal Reports by the Reports
Team. XXXXXX.

XXXXXXXX:

XXXXX Module: Pending the availability of XXXXXXX This will insure that modification to the test data and burden run schedules were successful.

Modifications to the XXXXXXX.

Strategic Systems Planning

Manager: XXXXXVolpe Initiator: XXXXX
XXXXX, 2002 to XXXXX, 2002

The generation of a XXXXXXX.

XXXXXXX XXXXXX: Efforts will be made to automate the XXXXXX. These changes will be demonstrated to the users during a simulated production run of the files. Once the related XXXXXXX a test XXXXX schedule run can be performed.

XXXXXXX: Pending the availability of XXXXX team resources, conversion activities XXXXX XXXXX. It is anticipated that answers to questions generated by the XXXXXX.

XXXXXX: The Volpe data requirements will be XXXXXX.

XXXXXXX: Data extract and validation activities will continue, as the XXXXX is refined. The design and implementation of the XXXXX will continue.

XXXXXX: XXXXX processes will continue to be refined and documented. The XXXXX will continue to be maintained. Support will continue to be provided to XXXXXX.

XXXXXXX: XXXXX meetings will continue as needed to refine the approach for implementing the XXXXX system.

XXXXX Web: Monitoring of Internet and Intranet sites to ensure availability and reliability will continue.

Intranet: Updates and maintenance activities will continue. XXXXXX.

Volpe Internet: Updates and maintenance activities will continue. XXXXXXX.

Special Interest Items/Issues

None.

Technical Topics

By
X
XXXXX to XXXXX

Migrating to the Windows 2000 Network Operating System

Table of Contents

IADPSS Monthly Progress Report
XXXXX, 2002 to XXXXX, 2002

Administrative Applications

Manager: XXXXX, Volpe Initiator: XXXXX
XXXXX, 2002 to XXXXX, 2002

Activities for Reporting Month

In the **Production and Maintenance Area**, general production support work continued with the following notable special maintenance activities.

Support was provided for processing of **XXXXXXXX** and **XXXXXX** data. This included resolving database and systems issues, and programming several ad-hoc reports. Normal batch processing of XXXXX and XXXX feeds into XXXX occurred.

The XXXXX database was purged of XXXXX codes.

The **XXXXX**, and **XXXXX Systems** experienced no problems during the period.

The **XXXXX System** users requested and received eight special reports.

The **XXXXX System** users had trouble accessing the data maintenance system. The problem was resolved by rebooting their PC.

The following notable **New Work Task** activities took place during the reporting month:

In the **XXXXX** area, weekly Work Group and bi-weekly Checkpoint meetings continued. Work on outstanding tasks associated with the contracts implementation continued. For more information on XXXXX specific issues, see the *Strategic Systems Planning* section of this report.

In the **XXXXX System** area, working sessions continued to complete the tasks associated with developing the XXXXX feed to XXXXX.

Analysis XXXX regarding the conversion of the XXXX database to the XXXXX platform and database structures.

For more information, see the *Strategic Systems Planning* section of this document.

In the **XXXXX** area, support was provided to the ongoing validation effort. Daily comparisons of XXXXX-Incremental data continued and were compiled, analyzed and presented to XXXXX.

The effort to add new fields to the data extracts for the **XXXX** area continued. These additional data fields are required to support the informational data requirements of XXXXX. An initiative to re-organize and finalize the data extracts for the XXXXX area was implemented. The enhanced format of the extracts will allow for more efficient management when the data staging area is released for XXXXX.

For more information, see the *Strategic Systems Planning* section of this document.

Activities for Next Month

During the next month, the following activities are planned in addition to performing routine production and maintenance support.

Support will be provided for month-end XXXXX and opening of XXXXX and the **XXXX** for the following month.

Normal maintenance activities are planned for **XXXXX,** and **XXXX Systems** for next period.

Normal maintenance activities are planned for the **XXXXX System**.

In the **XXXXX Systems** area, the subcommittee working sessions will continue to address the issues related to completing the XXXX task.

Programming of the XXXXX feed to XXXX will be tested.

Analysis of all legacy systems will continue to identify interfaces with the XXXX database.

Analysis will continue regarding the conversion of the XXXX database for the XXXXX project. The XXXX data migration will require XXXXXX.

In the **XXXX** area, the effort to monitor and analyze the XXXXX will continue. XXXXXX.

For more information about the XXXXX tasks, see the *Strategic Systems Planning* section of this document.

Special Interest Items/Issues

None.

AIS Security Program

Manager: XXXXX, Volpe Initiator: XXXXX
XXXXX, 2002 to XXXXX, 2002

Activities for Reporting Month

Data documentation for the XXXX Database including data XXXXX was created. In line with the data documentation, a newly created XXXXX was documented and proposed.

XXXXX.

The Center-wide XXXXXX

A complete listing of all outstanding operating system and software patches was supplied to XXXXX. XXXXX.

XXXXX efforts were once again conducted to provide security log reviews XXXXX.

A design was developed and an implementation process was performed to XXXXXX.

Remediation and configuration changes were applied to several servers as part of the continuing efforts to XXXXXX.

Several planning sessions were held regarding the XXXXXX

All XXXXX, XXXXX Web sites were reviewed for compliance to DOT web site regulations.

Discussions were initiated for security compliance, SSP preparation, and Certification efforts for XXXXX manager.

Planning meetings were held with the Applications Development team for the XXXX Project. Security Planning will be a XXX on-going component of this effort.

Several Vendor sessions were conducted XXXXX. These included presentations by Veritas, Symantec, RSA, Vericept and GTSI.

Approaches to meeting the Federal and DOT requirement for Awareness and Training were discussed. XXXXXX.

Research was conducted and results provided to XXXX regarding privacy statements on individual web pages.

XXXXXX.

Discussions were held regarding XXXXXXX.

Security issues related to password XXXXXX.

XXXXXXX.

The XXXXXX.

Windows 2000 Migration meetings were attended. XXXXXXXX.

A number of project/program meetings were attended for the purpose of providing input regarding security issues. XXXXXXXs.

Monitoring of log files continued on the firewall XXXXXXX.

XXXXXX.

XXXXXXX.

Discussions with Symantec provided updated information about their current firewall and security product offerings.

Activities for Next Month

Work will continue on updating all Volpe Center XXXXXXX.

ISS scans for vulnerabilities of both new systems being deployed and existing systems connected to the Volpe LAN will continue.

Reviews of current technology will be conducted to ensure a state-of-the-art capability is maintained for addressing security issues within the Volpe Center environment.

XXXXXXX

Efforts will be conducted to address any XXXXXXXX.

Reviews of logs will be conducted.

Ad hoc tasks will be addressed.

Special Interest Items/Issues

XXXXXXXX.

Facility Operations

Manager: XXXXXXXX, Volpe Initiator(s): XXXXXX
XXXXXX, 2002 to XXXXXX, 2002

Activities for Reporting Month

The following activities were completed or continue to be worked on by Facility Operations personnel during XXXXX.

The Data Entry team in **User Services** provided on-going support to XXXXX during this recording period.

The XXXXX group in User Services provided on-going support to the XXXXX during this recording period.

User Services provided Systems Administration support to the XXXXX during this recording period.

During XXXXX, the on-going support of storage management activities continued through general OpenVMS cluster disk maintenance and media management in the **XXXXX** area. The series of tasks in support of the Data Center OpenVMS cluster storage facility also included monitoring disk capacities to avoid device saturation, performing any/all monthly housekeeping on VCS media and performing general disk housekeeping on the XXXXX and XXXXX devices.

Through the month, the **OpenVMS Systems Managers** performed various systems administration tasks supporting XXXXXX.

The OpenVMS Systems Managers, in concert with the Systems Administration Team, performed UNIX system administration tasks that included XXXXX.

The OpenVMS Systems Managers completed work on the implementation of the XXXXX.

The OpenVMS Systems Managers, in concert with the Systems Administration Team, continued to work with XXXXX evaluation installation.

The OpenVMS Systems Managers, in concert with the Systems Administration Team, installed XXXXX.

The OpenVMS Systems Managers assisted Administrative Applications with modifications to the XXXXX.

The OpenVMS Systems Managers replaced and rebuilt the system disk on XXXXX.

The OpenVMS Systems Managers continued work on the XXXXXX provided the appropriate document to assist in determining what level of service could potentially be provided.

Oracle Initiatives

- XXXXX
- XXXXX
- XXXXXXXX

- During the month of XXXXX **the System Administration Team (UNIX and Windows NT/2000)** continued it's ongoing support and maintenance of all servers and associated services. XXXXXXXXX

Project Work:

Facility Operations

Manager: XXXXXXXX, Volpe Initiator(s): XXXXXX
XXXXXX, 2002 to XXXXXX, 2002

UNIX project work in the month of XXXXX included resolving a printer queue issue on the XXXXX.

The team worked with XXXXX.

Windows NT/2000 efforts included troubleshooting specific networking and connectivity problems with the *XXXXXX*.

Extensive support was provided to the XXXXXXX.

- **Network Engineering** completed the following XXXXX

- Network Engineering completed the following XXXXXX

During the UPS test late in the month of XXXXX, a failed UPS battery was identified in the Data Center; this UPS supports the XXXXXX

Network Engineering worked with XXXXX.

A XXXX Client has been XXXX.

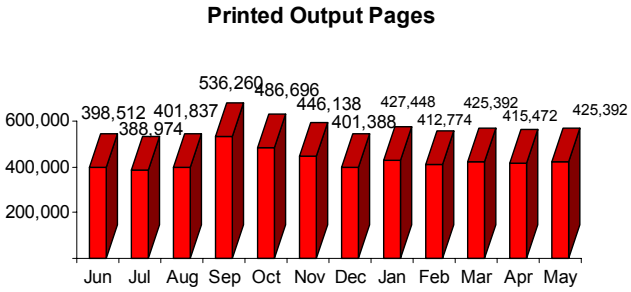
Activities for Next Month

- **Network Engineering** is preparing work on the following VPN Implementations for XXXXXX

XXXXX.

The **System Administration Team (UNIX and Windows NT/2000)** and the **OpenVMS Systems Managers** will continue their on-going support and maintenance of all Data Center servers as well as their associated services.

Other:



XXXXXXXXX.

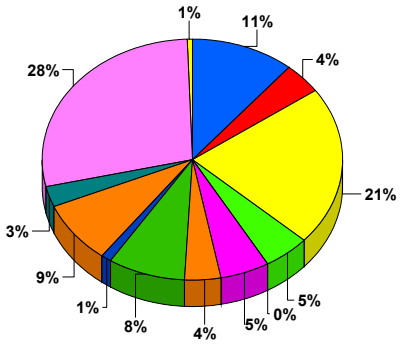
PC/Network Support

Manager: XXXXX, Volpe Initiator: XXXXXX
XXXXX, 2002 to XXXX, 2002

Activities for Reporting Month

The Exchange Mailbox totals for the current month are listed below:

	Additions	Deletions	Total
Mailboxes	28	3	1368
Remote Mail	3	0	289



Software	**260**
Hardware	**208**
EMAIL	**105**
Reset Password	**60**
Network	**37**
Others	**30**
Total Calls:	**700**

Exchange: Most of XXXXX was spent researching and planning the windows 2000 migration, specifically as it relates to the Exchange servers. XXXXXXX.

XXXXXXX.

Gateway Antivirus Activity Report

01-XXXXXXX-2002 00:00:00 - 01-XXXXXXX-2002 00:00:00

Message Summary

Messages Accepted	226306
Data Accepted (KB)	9181547
Messages Blocked By Subject	0
Messages Rejected	20
Messages Bounced	52
Messages Delivered	222745
Message Delivery Failures	11
Messages Completed	225793
Attachments Deleted	663

Infection Summary

Infections Ignored	0
Infections Repaired	5
Infections Deleted	124
Total	129
Infections Quarantined	0

Desktop: In XXXXX, the desktop technicians XXXXX.

In XXXXX, the desktop group continued to assist in the installation of IE XXXXX.

Special Interest Items/Issues

None.

Strategic Systems Planning

Manager: XXXXXVolpe Initiator: XXXXX
XXXXX, 2002 to XXXXX, 2002

Activities for Reporting Month

XXXXXX): The system was supported during the period without incident.

The XXXXX application re-hosting from the Open VMS operating system to the Tru64 Unix operating system continued. XXXXX.

The XXXX Area was used to populate an XXXX test database. XXXXX.

XXXXX: Several individual users with specific problems were assisted at their desktops. Several new accounts were created. XXXXX.

XXXXX: The development of a list of data errors that exist in XXXX is in-progress. This list will provide the basis for developing an XXXXX that will run daily and generate a diagnostic error report for use in correcting them.

Work also continued on generating required weekly XXXXX reports.

- XXXXX XXXXXXX

XXXXX XXXXX

XXXXX:

XXXXX Module: Meetings were held with XXXX to review the methods developed for production processing of the interface feeds to XXXXX. Several enhancements were made to the GUI XXXXX.

XXXXX Module: The XXXX instance was populated with sample data files. The actual steps for processing them have changed since the last time this was tested. All relevant processing scripts were accumulated and combined, and a production process solution was created for review by the XXXX group.

XXXXX: Members of the XXXXX team met and identified XXXX reports to use for conversion testing. XXXXX.

XXXXX: The XXXXXX team contacted the XXXXX staff responsible for XXXXX data extracts and is awaiting their response. No new activity occurred this month.

XXXXX: The XXXX software was received, and those portions XXXX were installed. XXXXX. XXXXXX software were installed successfully and the server was scheduled for the required ISS security scan.

Data validation activities continued, incorporating XXXXX along with the analysis of XXXXX Daily e-mails monitoring the differences between XXXXX-Incremental were analyzed and presented to XXXXXXX.

The planning effort for the design and implementation of a XXXX was completed.

XXXXXX: The XXXX Requirements and the XXXXXX Requirements were finalized and await submission to the XXXX team once the new XXXX plan is established.

Strategic Systems Planning

Manager: XXXXXVolpe Initiator: XXXXX
XXXXX, 2002 to XXXXX, 2002

The XXXXX processes were determined, refined, and draft documentation was generated. This included XXXXXXX.

XXXXXX: Technical and planning meetings were temporarily suspended.

XXXX Web: Both the XXXXX sites were monitored to ensure availability and reliability. Feedback for both was gathered and responses coordinated.

The evaluation of XXXXX for use with the Intranet and Internet continued.

Planning continued for an upgrade of the current XXXXXX servers.

XXXXX team members continued to participate in meetings related XXXXXX.

XXXXX: The XXXXXX was updated with new XXXXXXXXX.

XXXXXXs were launched on the Intranet.

Information on Technical Topics is in XXXXXXX.

A "XXXXXXXX.

XXXXXXX.

XXXXXXXXX.

Volpe Internet: Updates were made to several sections, including the XXXXXX. Work continued on XXXXX to meet Section 508 standards.

XXXXXXX.

Other Internet: The XXXXXX.

XXXXX team members supported a short-term effort related to the XXXXXXX

Activities for Next Month

XXXXXXX): Work will continue to establish additional fiscal year data in the test database XXXXXXX.

XXXXXX: Assistance to Desktop Support and User Administration will continue. Research will continue regarding the XXXXXX.

XXXXXX): Selected XXXXX reports will be identified for development.

Work will continue to resolve the XXXXXX.

XXXXXreports were selected for conversion to Crystal Reports by the Reports
Team. XXXXXX.

XXXXXXXX:

XXXXX Module: Pending the availability of XXXXXXX This will insure that modification to the test data and burden run schedules were successful.

Modifications to the XXXXXXX.

Strategic Systems Planning

Manager: XXXXXVolpe Initiator: XXXXX
XXXXX, 2002 to XXXXX, 2002

The generation of a XXXXXXX.

XXXXXXX XXXXXX: Efforts will be made to automate the XXXXXX. These changes will be demonstrated to the users during a simulated production run of the files. Once the related XXXXXXX a test XXXXX schedule run can be performed.

XXXXXXX: Pending the availability of XXXXX team resources, conversion activities XXXXX XXXXX. It is anticipated that answers to questions generated by the XXXXXX.

XXXXXX: The Volpe data requirements will be XXXXXX.

XXXXXXX: Data extract and validation activities will continue, as the XXXXX is refined. The design and implementation of the XXXXX will continue.

XXXXXX: XXXXX processes will continue to be refined and documented. The XXXXX will continue to be maintained. Support will continue to be provided to XXXXXX.

XXXXXXX: XXXXX meetings will continue as needed to refine the approach for implementing the XXXXX system.

XXXXX Web: Monitoring of Internet and Intranet sites to ensure availability and reliability will continue.

Intranet: Updates and maintenance activities will continue. XXXXXX.

Volpe Internet: Updates and maintenance activities will continue. XXXXXXX.

Special Interest Items/Issues

None.

Technical Topics

By
X
XXXXX to XXXXX

Migrating to the Windows 2000 Network Operating System

Table of Contents

IADPSS Monthly Progress Report
XXXXX, 2002 to XXXXXX, 2002

Administrative Applications

Manager: XXXXXXX, Volpe Initiator: XXXXXXX
XXXXXXX, 2002 to XXXXXXX, 2002

Activities for Reporting Month

In the **Production and Maintenance Area**, general production support work continued with the following notable special maintenance activities.

Support was provided for processing of **XXXXXXX** data. This included extensive analysis regarding XXXXXX, programming several ad-hoc reports against the XXXXX file, programming several XXXXXX, and programming a report modification to XXXXXX. Normal batch processing of XXXX and XXXXX into XXXXX occurred.

The **XXXXX**, and **XXXXX Systems** experienced no problems during the period.

The **XXXXX System** was modified to eliminate the production of two reports and to place a new report into production. The users requested and received two special reports.

The **XXXXXX System** users requested and received a special request report of XXXXXX.

The following notable **New Work Task** activities took place during the reporting month:

In the **XXXXX** area, weekly Work Group and bi-weekly Checkpoint meetings continued. Work on outstanding tasks associated with the XXXXXX continued. For more information on XXXXXX specific issues, see the *Strategic Systems Planning* section of this report.

In the **XXXXX** area, working sessions continued to complete the tasks associated with developing the XXXX to XXXX.

A telcon occurred mid-month regarding the conversion of the XXXXXX database to the XXXXX platform and database structures.

For more information, see the *Strategic Systems Planning* section of this document.

In the **XXXX** area, support was provided to the ongoing validation effort by daily monitoring of the XXXXXX.

The **XXXXX** extract program was modified and tested to allow for separate PRPO XXXXX. XXXX mail was installed and tested by SysAdmin on XXXXX and the extract programs were modified to provide XXXXXX. In addition, minor changes were made to some of the fields in the data extract as required by XXXXX.

For more information, see the *Strategic Systems Planning* section of this document.

Activities for Next Month

During the next month, the following activities are planned in addition to performing routine production and maintenance support.

Support will be provided for month-end closing of the **XXXX** and opening of XXXX and the **XXXX** for the following month.

Normal maintenance activities are planned for **XXXXX,** and **XXXX Systems** for next period.

Normal maintenance activities are planned for the **XXXX System**.

In the **XXXX Systems** area, the subcommittee working sessions will continue to address the issues related to completing the XXXX task.

Programming of the XXXX will be tested.

Analysis of all XXXX systems will continue to identify interfaces with the XXXX database.

Analysis will continue regarding the conversion of the XXXX database for the XXXX project. The XXXX data migration will require extensive teamwork among XXXX.

In the **XXXX** area, the effort to monitor and analyze the data in the XXXX will continue. In addition, programming and analytical support will continue to be provided in support of **XXXX** area activities.

For more information about the XXXXX tasks, see the *Strategic Systems Planning* section of this document.

Special Interest Items/Issues

None.

AIS Security Program

Manager: XXXXX, Volpe Initiator: XXXXX
XXXXX, 2002 to XXXXX, 2002

Activities for Reporting Month

Reviewed and documented all web sites hosted at the Volpe Center for compliance to DOT regulations.

The Center-wide vulnerability scanning effort focused on XXXXX XXXXXX.

Provided recommendation to XXXXteam for implementing XXXX for use in all data transfers within XXXX.

XXXXX.

Presented a secure design for hosting XXXXX.

Remediation and configuration changes were applied to several servers XXXXXXX.

Met with XXXX to discuss optimum way to implement XXXX certificates over XXXXX.

The XXXXXX was expanded and further additions to the Plan were incorporated.

XXXXX were updated and revised. One XXXXX remains to be completed.

Provided research to XXXXX regarding XXXX on individual web pages.

The final status report for the XXXX project management.

XXXXX services are being researched and analyzed. A recommendation for use at the Volpe Center was provided to management.

XXXXXX was reviewed with System Administration.

Work commenced with the XXXXXe group in ensuring the development of the XXXX is done according to DOT certification and accreditation standards.

Activities associated with the XXXXBackup project continued.

Participated in XXXXX Migration meetings. XXXXX.

A number of XXXX meetings were attended for the purpose of providing input regarding security issues. These included XXXXX programs.

Monitoring of log files continued on the XXXXX.

Several ad hoc activities were conducted XXXXXX.

Discussions were XXXX with XXXXX services.

Activities for Next Month

Work will continue on the XXXXX Programs as appropriate.

Work will continue on updating all XXXX.

XXXXX of both new systems being deployed and existing systems connected to the Volpe LAN will continue.

Reviews of current technology will be conducted to ensure a state-of-the-art capability is XXXXX environment.

Work will continue on the XXXX Project. That project will be more tightly intertwined with XXXX project.

Research continues with regards to upgrading the XXXXXX.

XXXXXXX.

Reviews of logs will be conducted.

Ad hoc tasks will be addressed.

Special Interest Items/Issues

XXXXXXXXX.

Facility Operations

Manager: XXXXXXXX, Volpe Initiator(s): XXXXXX
XXXXX, 2002 to XXXXXXX, 2002

Activities for Reporting Month

The following activities were completed or continue to be worked on by Facility Operations personnel during XXXXXX 2002.

The XXXX team in **User Services** provided on-going support to the XXXXXX during this recording period.

The XXXXX provided on-going support to the Airworthiness project, XXXXX during this recording period.

User Services provided Systems Administration support to the XXXXX during this recording period.

During XXXX 2002, the on-going support of XXX activities continued through general OpenVMS cluster disk maintenance and media management in the **XXXX** area. The series of tasks in support of the XXXX facility also included monitoring disk capacities to avoid device saturation, performing any/all monthly housekeeping on XXX and performing general disk XXXXXX.

Through the month, the **OpenVMS Systems Managers** performed various systems administration tasks supporting the XXXXXX.

The OpenVMS Systems Managers, in concert with the Systems Administration Team, performed UNIX system administration tasks that included XXXXXX.

The OpenVMS Systems Managers upgraded the XXXXXX.

The OpenVMS Systems Managers enabled e-mail communications from XXXXX.

The OpenVMS Systems Managers Managers wrote a XXXXX.

Due to Developer changes on the XXXXX, the OpenVMS Systems Managers had to request a XXXXX.

The OpenVMS Systems Managers added XXXXX Gigabyte disk drives and an additional XXXX Channel card to XXXXXX.

Oracle Initiatives

- XXXXX.
- XXXXXX.
- During the month of XXXX, the **System Administration Team XXXX** continued it's ongoing support and maintenance of all servers and associated services. This included XXXXXX

The Systems Administration Team met with the XXXXXXX

Project Work:

UNIX project work in the month of XXXX included the migration of the XXXXX.

UNIX patches and XXXXX were planned.

Facility Operations

Manager: XXXXXXXX, Volpe Initiator(s): XXXXXX
XXXXX, 2002 to XXXXXXX, 2002

The Systems Administration Team assisted with efforts to migrate XXXXX.

The Systems Administration Team built the XXXXX

Windows NT/2000 efforts for the month included XXXXX.

XXXXXX.

The Systems Administration Team supported the XXXXXXXX.

Volpe XXXXXX.

XXXXXXXX

Network Engineering XXXXXXX.

Network Engineering completed the following XXXXXXXXX

Network Engineering completed all required network infrastructure installations and configurations XXXXXXXX.

Network Engineering completed the layout and installation of XXXXXXXX.

Network Engineering had the primary responsibility for XXXXXXXX

The XXXXXXXX.

Network Engineering provided network support for XXXXXX.

Network Engineering tested VPN connectivity XXXXXX.

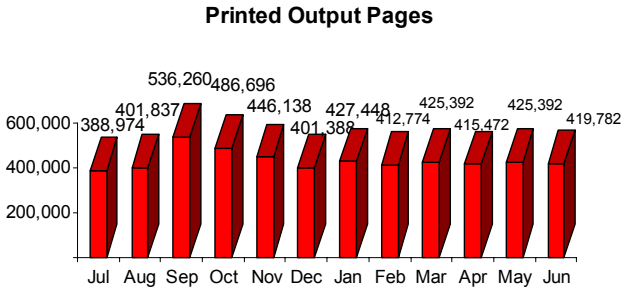
Activities for Next Month

Network Engineering is in process with the XXXXX

Network Engineering is in the process of setting up a XXXXXXX

The **System Administration Team XXXXX** and the **OpenVMS Systems Managers** will continue their on-going support and maintenance of all Data Center servers as well as their associated services.

Other:



Special Interest Items / Issues

PC/Network Support

Manager: XXXXXX, Volpe Initiator: XXXXXX
XXXXXX, 2002 to XXXXXX, 2002

Activities for Reporting Month

The Exchange Mailbox totals for the current month are listed below:

	Additions	Deletions	Total
Mailboxes	28	5	1391
Remote Mail	4	0	293

Data Accepted (KB)	8123494
Messages Blocked By Subject	0
Messages Rejected	28
Messages Bounced	41
Messages Delivered	209829
Message Delivery Failures	13
Messages Completed	191126
Attachments Deleted	1066

All Calls Received between XXXX and XXXXX

xxxxxxxxxx

Email	**73**
General Inquiry	**24**
Hardware-Instl	**32**
Hardware-Probs	**29**
Mainframe Issue	**1**
Move Add Change	**15**
Network	**39**
New Account	**27**
New PC Install	**50**
Printing	**34**
Reset Password	**48**
Server Issues	**17**
Software	**110**
Grand Total of Call ID:	**530**

Exchange: XXXX X

Desktop: In XXXXX, the desktop technicians XXXXXX.

In XXXXX, the desktop group continued with the installation of IE XXXXXXX.

Heat: The Heat database has been upgraded to version 6.4. XXXXXXX.

Special Interest Items/Issues

None.

Gateway Antivirus Activity Report

XXXXXX0:00:00 - XXXXXXXX 00:00:00

Message Summary

Messages Accepted	191969

Strategic Systems Planning

Manager: XXXXXX Volpe Initiator: XXXXXX
XXXXXX, 2002 to XXXXXX, 2002

Activities for Reporting Month

XXXXXX: The system was supported during the period without incident. One new user was added during the period.

The XXXX from the Open VMS operating system to the Tru64 Unix operating system continued. XXXXXX.

The XXXXX was used to populate an XXXXXX. Validation of data continued with success.

XXXXX: Several individual users with specific problems were assisted at their desktops. Several new accounts were created.

XXXXXXXX: The development of a list of data XXXXX is in-progress. This list will provide the basis for developing XXXXXXX.

- XXXXXXX
- XXXXXXX
  - XXXXXXX

- Four Additional Crystal Reports are being developed by the group:
  - XXXXXXXXX

XXXXXXX

XXXXXXX:

XXXXXX Module: Enhancements were made to the interface file preview scripts based on feedback from the XXXXXX group. A review of those enhancements will be performed during the next round of interface testing. XXXXXX. Request for further support was submitted.

XXXXXX Module: XXXXX.

XXXXXXX: Members of the XXXXX team met and made progress towards finalizing some outstanding issues.

A teleconference occurred with XXXXXX. No further discussions occurred during the month.

XXXXXX: The XXXXX team contacted the XXXXXX. No new activity occurred this month.

XXXXXX: The XXXXXXX. Following the completion of XXXXX design phase, the XXXX development was begun.

XXXXXXX The XXXX Area processes were further refined, tested, and documented. XXXXXXX

XXXXX: Technical and planning meetings were temporarily suspended.

XXXX Web: Both the Internet and Intranet sites were monitored to ensure availability and reliability. Feedback for both was gathered and responses coordinated.

XXXXXXX

Web development team members continued to participate in meetings related to XXXXXXX.

Strategic Systems Planning

Manager: XXXXXX Volpe Initiator: XXXXXX
XXXXXX, 2002 to XXXXXX, 2002

XXXX: The XXXX was updated with new content, and routine maintenance tasks were performed. XXXXXXX.

Discussions are under way to remove out-dated and redundant items from Shared Folders, XXXXXXXX.

Information on Technical Topics is in the final stages of XXXXXXX

XXXX.

XXXXX.

XXXXXX.

Internet: Updates were made to several sections, including XXXXX.

Additional subsites are under development.

New privacy policy pages were developed and posted to meet requirements issued by DOT.

Other Internet: XXXXXXX.

New privacy policy pages were developed and posted to meet requirements issued by DOT.

Activities for Next Month

XXXXXXX Work will continue to establish additional XXXX data in the test database derived from data in the XXXXX. Activity pertaining to the move of XXXXX will also continue.

XXXXX: Assistance to Desktop Support and User Administration will continue. Research will continue regarding the XXXXX issue.

XXXXX: Selected XXXXX reports will be identified for development.

Work will continue to resolve the XXXXX identified.

XXXXX.

XXXXXX.

Strategic Systems Planning

Manager: XXXXXX Volpe Initiator: XXXXXX
XXXXXX, 2002 to XXXXXX, 2002

XXXXXX

XXXXX:

XXXXX Module: On-site XXXXX support from XXXXX is anticipated to be available. A fresh set of test data will be prepared and the procedures used to process a new interface file will be documented.

XXXXXX Module: The XXXXXXX

XXXXXX: Pending the availability of XXXXX Implementation team resources, conversion activities could resume.

XXXXXXX: The XXXXXX requirements will be reviewed with XXXXXX Implementation team pending their availability. XXXXXX.

XXXXXXXX: XXXXX activities will continue as the XXXXX is refined. The design and implementation of the XXXXX will continue.

IXXXXXX: XXXXX Area processes will continue to be refined, tested and documented. XXXXXXX.

XXXXXX: XXXXX meetings will continue as needed to refine the approach for implementing XXXXX capability.

XXXX Web: Monitoring of Internet and Intranet sites to ensure availability and reliability will continue.

Intranet: Updates and maintenance activities will continue. XXXXXXX.

Volpe Internet: Updates and maintenance activities will continue. XXXXXXX.

Special Interest Items/Issues

None.

Technical Topics

XXXXXX, 2002 to XXXXXXXX, 2002

NO TECHNICAL TOPIC FOR JUNE

Doc Ref Number: **Revision Number:**

Revision Date:

Procedure Type: **Page Number:**

TITLE OF STANDARD OPERATING PROCEDURES

1. **Identify the Stakeholders who should be involved in the approval process.**

2. **Identify and cite the specific any applicable standards, guidelines, policies or orders that apply to this procedure.**

3. **Summary Description of SOP purpose/results.**

4. **Responsible organization for SOP and hierarchy of actions.**

5. **Detailed description of SOP**

    - **Milestones**
    - **Timelines**
    - **Periodicity**
    - **Measures of Performance**

6. **Reporting Mechanism for SOP**

7. **Contact name and information for SOP**

8. **Checklist for SOP including signature blocks**